**Question 1.**

Till last week, we had talked about computing the solution to the least squares problem, where we have taken different cases of A. These can be easily extended, as was seen in the constrained least squares and generalized Tikhonov regularization procedures. But these mainly dealt with solving a system of positive definite equations. We will need to look into matrix properties to figure out the solutions to a general system of equations. These involve the effective factorization of matrices into matrices that are easier to solve. Diagonal, orthogonal and triangular systems are the backbones of matrix factorization. Computation with these types of matrices will yield faster results as complexity is less. Using these as the basic concepts, we look into other complex factorization methods. LU factorization converts any symmetric matrix into a lower triangular matrix with an upper trianglular matrix. This is basically done by recording the steps of Gaussian Elimination while reducing a matrix. The Cholesky decomposition is simply a particular case of the LU decomposition for symmetric positive definite matrices. It is also an iterative process and it can be computed faster than a general LU decomposition and is easier to stabilize. A matrix can be factored into a product of an orthogonal and an upper triangular matrix by the QR factorization.

To deal with the calculation of eigen values and eigen vectors for a symmetric positive semi-definite matrix, we discussed two main procedures. These are the power iterations method and QR iterations method. Power iterations were used to find the largest eigen value and the corresponding eigen vector, but QR iterations can be used to calculate all the eigen values and their eigen vectors. Problems of machine learning in the real world involve a lot of data and thus, we definitely need certain types of structured symmetric matrices for easier solving of $Ax = b$. The structure of identity+low rank, circulant, Toeplitz, or banded matrices will allow us to do a solve in much better than $O(N^3)$ operations. The next week, we shall discuss two different iterative algorithms for computing the solution to $Ax = b$, when A is symmetric and positive definite. For large dimensional A, it might not be feasible to store A. These algorithms do not hold A in memory. Thus, given x as an input, these algorithms will compute Ax. We shall look more into this during the coming week.