R Notebook

Problem #1

Proof and Demonstration

We have a matrix size m * n. The product of their square A^TA noted as n * n and AA^T as matrix of m * n. To show $A^TA \neq AA^T$, it means that n$n\ is\ not\ equal\ to\ m$m unless n = m which mean A is a square. Therefore, for non-square matrices,$A^TA \neq AA^T$ have different dimensions and cannot be equal.

$A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$.

The product of AA^T will be: $\begin{bmatrix} a \times a + b \times b & a \times c + b \times d \\ c \times a + d \times b & c \times c + d \times d \end{bmatrix}$.

The product of A^TA will be: $\begin{bmatrix} a \times a + c \times c & a \times b + c \times d \\ a \times b + c \times d & b \times b + d \times d \end{bmatrix}$.

```
A <- matrix(seq(from = 2,8, by =2), nrow = 2)
A
```

```
##      [,1] [,2]
## [1,]    2    6
## [2,]    4    8
```

```
# creating AT
A_T <- t(A)
A_T
```

```
##      [,1] [,2]
## [1,]    2    4
## [2,]    6    8
```

```
# Creaing A^TA
AT_A <- t(A)%*%A
AT_A
```

```
##      [,1] [,2]
## [1,]   20   44
## [2,]   44  100
```

```
#Creating AA^T
AA_T <- A%*%t(A)
AA_T
```

```
##      [,1] [,2]
## [1,]   40   56
## [2,]   56   80
```

Now we can argue that $A^TA \neq AA^T$. $A^TA = AA^T$ only when the matrix is orthogonal matrix and its transpose is equal to its inverse matrix. We have a matrix :

A $= A^T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$.

$A^T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$.

$AA^T = \begin{bmatrix} (-1) \times (-1) + 0 \times 0 & 0 \times 0 + 0 \times 1 \\ 0 \times (-1) + (1) \times 0 & 0 \times 0 + 1 \times 1 \end{bmatrix}$.

$AA^T = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$.

Problem #2

Matrix factorization is a very important problem. There are supercomputers built just to do matrix factorizations. Every second you are on an airplane, matrices are being factorized. Radars that track flights use a technique called Kalman filtering. At the heart of Kalman Filtering is a Matrix Factorization operation. Kalman Filters are solving linear systems of equations when they track your flight using radars.

Function that factorize a square matrix A LU and LDU:

```r
# Function that factorize the matrix
lu_decomposition <- function(A){
  #Check for valid input
  if(nrow(A) != ncol(A)){
    stop("Input must be square matrix.")
  }
  else {
    n <- nrow(A)
    L <- diag(n)
    U <- A
    for (i in 1:(n - 1)) {
      for (j in (i + 1):n) {
        # get multipliers
        L[j, i] <- U[j, i] / U[i, i]
        # pivots and multiplication
        U[j, ]  <- U[j, ] - L[j, i] * U[i, ]
      }
    }

    #Get results
    LU <- list("L" = L, "U" = U)
  }
}

# Example usage:
b <- matrix(c(2, -1, -2, 4, 1, 2, -4, 6, 3), 3, 3)
result <- lu_decomposition(A)
```

```r
result
```

```
## $L
##      [,1] [,2]
## [1,]    1    0
## [2,]    2    1
##
## $U
##      [,1] [,2]
## [1,]    2    6
## [2,]    0   -4
```

```r
L <- result[[1]]
U <- result[[2]]



print(lu_decomposition(b)$L %*% lu_decomposition(b)$U  == b)
```

```
##      [,1] [,2] [,3]
## [1,] TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE
```