DATA605 - Assignment 2

Nick Oliver

Assignment 2

Problem set 1

(1) Show that

$$A^T A \neq AA^T$$

in general. (Proof and demonstration.)

(2) For a special type of square matrix A, we get AT A = AAT . Under what conditions could this be true? (Hint: The Identity matrix I is an example of such a matrix). Please typeset your response using LaTeX mode in RStudio.

(1)

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} A^T = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} A * A^T = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} A^T * A = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} A * A^T \neq A^T * A$$

```
A = matrix(c(0,1,0,1), byrow = TRUE, nrow = 2)
A
```

```
##      [,1] [,2]
## [1,]    0    1
## [2,]    0    1
```

```
tA = t(A)
tA
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    1    1
```

```
A %*% tA
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    1    1
```

```
tA %*% A
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    2
```

(2)

A "diagonal" matrix follows the property

$$A * A^T = A^T * A$$

A diagonal matrix is a square matrix with 0s outside of the main diagonal. The identity matrix is a trivial example of a diagonal matrix.

$$A = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} A^T = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} A * A^T = \begin{bmatrix} 36 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 16 \end{bmatrix} A^T * A = \begin{bmatrix} 36 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 16 \end{bmatrix} A * A^T = A^T * A$$

Problem set 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.

Write an R function to factorize a square matrix A into LU or LDU, whichever you prefer. You don't have to worry about permuting rows of A and you can assume that A is less than 5x5, if you need to hard-code any variables in your code. If you doing the entire assignment in R, then please submit only one markdown document for both the problems.

Function Definition

```
#Doolittle Algorithm
factorizeIntoLU <- function(A) {
  size <- dim(A)[1] #assuming square matrix

  L <- diag(0,size)
  U <- diag(0,size)

  for (i in seq(1,size)) {
    for (k in seq(i, size)) {
            acc <- 0;
            for (j in seq(1,i)) {
                acc <- acc + (L[i, j] * U[j, k]);
            }
            U[i, k] <- A[i, k] - acc;
        }

        for (k in seq(i,size)){
            if (i == k){
                L[i, i] <- 1
            }
            else {
                acc <- 0;
                for (j in seq(1,i)){
                    acc <- acc + (L[k, j] * U[j, i]);
                }
                L[k, i] <- (A[k, i] - acc) / U[i, i];
            }
        }
  }
print('L')
print(L)
print('U')
```

```
print(U)
}
```

```
A <- matrix(c(1,2,4,3,8,14,2,6,13),nrow = 3, byrow=TRUE)
A
```

Test Case 1

```
##      [,1] [,2] [,3]
## [1,]    1    2    4
## [2,]    3    8   14
## [3,]    2    6   13
```

```
factorizeIntoLU(A)
```

```
## [1] "L"
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    3    1    0
## [3,]    2    1    1
## [1] "U"
##      [,1] [,2] [,3]
## [1,]    1    2    4
## [2,]    0    2    2
## [3,]    0    0    3
```

```
matrixcalc::lu.decomposition(A)
```

Comparison With Library (1)

```
## $L
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    3    1    0
## [3,]    2    1    1
##
## $U
##      [,1] [,2] [,3]
## [1,]    1    2    4
## [2,]    0    2    2
## [3,]    0    0    3
```

```
A <- matrix(c(1,4,-3,-2,8,5,3,4,7),nrow = 3, byrow=TRUE)
A
```

Test Case 2

```
##      [,1] [,2] [,3]
## [1,]    1    4   -3
## [2,]   -2    8    5
## [3,]    3    4    7
```

```
factorizeIntoLU(A)
```

```
## [1] "L"
##      [,1] [,2] [,3]
## [1,]    1  0.0    0
## [2,]   -2  1.0    0
## [3,]    3 -0.5    1
## [1] "U"
##      [,1] [,2] [,3]
## [1,]    1    4 -3.0
## [2,]    0   16 -1.0
## [3,]    0    0 15.5
```

```
matrixcalc::lu.decomposition(A)
```

Comparison With Library (2)

```
## $L
##      [,1] [,2] [,3]
## [1,]    1  0.0    0
## [2,]   -2  1.0    0
## [3,]    3 -0.5    1
##
## $U
##      [,1] [,2] [,3]
## [1,]    1    4 -3.0
## [2,]    0   16 -1.0
## [3,]    0    0 15.5
```