BASIC MATRIX OPERATIONS AND PROPERTIES

1. Basic Matrix properties

Matrices are rectangular arrays with m rows and n columns. Matrices can be added together if their shapes are the same. That is, if they have the same number of rows and columns. To add matrices together, you simply operate on them element-by-element. You can see this by creating two matrices in R. For example:

```
A <- matrix(seq(from=1,to=6), nrow=2, byrow=T)
B <- matrix(seq(from=12,to=7), nrow=2)
A + B
```

You should see something like:

```
     [,1] [,2] [,3]
[1,]   13   12   11
[2,]   15   14   13
```

You'll see that as long as the matrices are compatible, R will add them element by element and produce a result matrix of the same shape.

To multiply matrices, the number of columns of the matrix on the left has to match the number of rows of the matrix on the right. That is, we can multiply $A \times B$, if and only if A is an mxn matrix and B is an nxk matrix. The product AB will be an $m \times k$ matrix. Each entry in this $m \times k$ matrix is a dot-product between one row of A and one column of B. Since each row of A has n elements and each column of B has n elements, every dot-product is well-defined. AB_{ij}, or the element at row i and column j in matrix AB is the dot-product between row i of A and column j of B.

Please test this out by multiplying two matrices in R by typing $A\%{*}\%t(B)$ and examining the result. It is crucial that you understand this row-column dot-product. Note that we used $t(B)$ to take the transpose of B in order to make our example matrices compatible with each other for multiplication. Please also try $t(A)\% * \%B$ and see what you get. These two examples should give you a good understanding of what is happening when two matrices are getting multiplied.

2. Laws of Matrices

Matrices follow these basic laws below:

$$
\begin{aligned}
A + B &= B + A \\
c(A + B) &= cA + cB \\
A + (B + C) &= (A + B) + C \\
C(A + B) &= CA + CB \\
(A + B)C &= AC + BC \\
AB &\neq BA \\
(AB)C &= A(BC)
\end{aligned}
$$

where c is a scalar number.

3. Square matrices and inverses

When m and n are equal, the matrix is a square matrix. When you have an $m \times m$ square matrix where all the (i, i) elements are 1 and all other elements are zero, it is called an Identity matrix. Below is a 3×3 identity matrix.

$$
I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

Identity matrices are special because $IA = A = AI$ for any compatible matrix A.

Identity matrices are a special form of a diagonal matrix, where all the diagonal elements are 1. In general, a diagonal matrix only has non-zero elements along the diagonal and zeros everywhere else.

Some square matrices have complement matrices which when multiplied together, produce Identity matrices. That is $AB = I$. If there exists a matrix B such that $AB = I$ and $BA = I$ then it is called the *inverse* of A and denoted as A^{-1}. Not all matrices are invertible. If a matrix is invertible, then for any non-zero x, $Ax \neq 0$. When we apply the elimination procedure, we get exactly n pivots (n being the dimension of the square matrix A) with an invertible matrix.

If two matrices A and B are invertible, then AB is also invertible and its inverse is $B^{-1}A^{-1}$.

4. Transpose of a matrix

For any matrix, we can define a special operation called the transpose. Given a matrix A, we define its transpose, A^T as a matrix whose rows are the columns of A and vice versa. So, if A is $m \times n$, A^T is $n \times m$.

When you have two vectors x and y, we can consider them as essentially two $n \times 1$ matrices. From this perspective, the dot-product between x and y is simply a matrix multiplication between x^T and y. x^T is $1 \times n$ and y is $n \times 1$ producing a 1×1 product. Therefore $x \cdot y$ is also be written as $x^T y$.

5. Factorization of A into LU

We'll see that many of the operations we'll do on matrices are factorizing them into multiple matrices, typically 2 or 3 matrices. In fact, the Elimination procedure from Week 1 is simply a step towards factorizing A into a product of two matrices: $A = LU$ where U is the matrix that we get at the end of the elimination procedure, it is the Upper Triangular matrix. L is a Lower Triangular matrix and you'll see that the entries of L are the multipliers that we applied to subtract one row from the other.

If we start with the matrix A

$$A = \begin{bmatrix} 2 & 1 \\ 6 & 8 \end{bmatrix} \tag{1}$$

We see that we need to subtract 3 times row 1 from row 2 to make the element 6 go to zero. Doing this using matrices, we get

$$\begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 0 & 5 \end{bmatrix} \tag{2}$$

The last matrix is our U matrix. Now, going back from U to A, we need the inverse of the elimination matrix.

$$\begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 6 & 8 \end{bmatrix} \tag{3}$$

In the above equation, we see that the A is factorized into two terms - the upper triangular and the lower triangular matrix.

In the 3×3 case, we have more elimination operations. We need to eliminate one element from row 2 and two elements from row 3. Assuming that there are no row exchanges needed, we will first multiply A with E_{21}, the elimination matrix that zeros out the element at position $(2, 1)$. We also now need to zero out elements $(3, 1)$ and $(3, 2)$. For this, we'll employ two more matrices E_{31} and E_{32}. Therefore, we get $(E_{32}E_{31}E_{21})A = U$. Reversing the procedure, we have $A = (E_{21}^{-1}E_{31}^{-1}E_{32}^{-1})U$. We'll call $(E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}) = L$, giving us $A = LU$.

Note here that the L matrix contains the multipliers l_{ij} that is needed to subtract row j from row i to eliminate the jth element from row i. The elimination matrices were primarily for illustration and we don't need to explicitly calculate them as long as we keep good book-keeping of the multipliers and the resulting U matrix. We can simply form the L matrix from an Identity matrix and multipliers located below the diagonal. This is simply one procedure to factorize A into LU. In general, LU decomposition is not unique and you can find several L and U matrices that satisfy.

Let's work out a more elaborate example in R. Let's start with the following 3×3 matrix A:

```
> A <- matrix(c(1,2,3,1,1,1,2,0,1),nrow=3)
```

```
> A
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    2    1    0
[3,]    3    1    1
```

In order to eliminate the 2 in the 2nd row, we can set up the following elimination matrix E21:

```
> E21 = matrix(c(1,-2,0,0,1,0,0,0,1),nrow=3)
> E21
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]   -2    1    0
[3,]    0    0    1
> E21 %*% A
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    0   -1   -4
[3,]    3    1    1
```

To eliminate row 3, we can choose E32 to be:

```
> E31 = matrix(c(1,0,-3,0,1,0,0,0,1),nrow=3)
> E31
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]   -3    0    1
> E31 %*% E21 %*% A
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    0   -1   -4
[3,]    0   -2   -5
```

Notice that the pivot in row two, $a_{22} = -1$. This leads us to choose the following elimination matrix E32:

```
> E32 = matrix(c(1,0,0,0,1,-2,0,0,1),nrow=3)
> E32
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0   -2    1
> U <- E32 %*% E31 %*% E21 %*% A
```

```
> U
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    0   -1   -4
[3,]    0    0    3
```

And that gives us the desired U matrix. We know that we can compute L as the inverse of the product of the elimination matrices. That is, $L = (E32 \times E31 \times E21)^{-1} = E21^{-1} \times E31^{-1} \times E32^{-1}$.

```
> L <- solve(E21) %*% solve(E31) %*% solve(E32)
> L
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    2    1    0
[3,]    3    2    1
```

Note that we used the *solve* function in R to get the inverse of the matrices. Finally, we see that $A = LU$.

```
> L %*% U
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    2    1    0
[3,]    3    1    1
> A
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    2    1    0
[3,]    3    1    1
> (L %*% U == A)
     [,1] [,2] [,3]
[1,] TRUE TRUE TRUE
[2,] TRUE TRUE TRUE
[3,] TRUE TRUE TRUE
```

We note that L matrix has ones along its diagonal, whereas the U matrix has pivots along its diagonal. This can be easily rectified by modifying U by pulling out the pivots into a diagonal matrix by itself and dividing U by the pivots to produce $A = LDU^{'}$ where $U = DU^{'}$

Matrix A can be factorized into LU or into $LDU^{'}$ to make L and $U^{'}$ both have 1s along the diagonal. Whenever you see 3 factors, with a diagonal D, it is understood that the upper triangular matrix has been factorized into two matrices so that it has 1s along its diagonal elements.

This week's assignment will require you to write code to perform LU decomposition on matrices. While these functions are built-in R and other math software, it is a useful exercise to understand how such an operation might be done behind the scenes.