1. $A = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix}, B = \begin{pmatrix} 2 & 1 & 7 & 6 \\ -1 & 8 & -3 & 4 \end{pmatrix}, x = (1,3,5,-1)^T$에 대하여 다음의 식을 확인하시오.

   (a) $(A + A^T)B = AB + A^TB$

   (b) $\mathrm{tr}(B^TB) = \mathrm{tr}(BB^T)$

   (c) $x^TB^TBx = \mathrm{tr}(B^TBxx^T)$

```
> #Q1
>
> A = matrix(c(1,0,2,-1), nrow = 2)
> B = matrix(c(2,-1,1,8,7,-3,6,4), nrow = 2)
> x = c(1,3,5,-1)
> A
     [,1] [,2]
[1,]    1    2
[2,]    0   -1
> B
     [,1] [,2] [,3] [,4]
[1,]    2    1    7    6
[2,]   -1    8   -3    4
> x
[1]  1  3  5 -1
> #(a)
> (A + t(A)) %*% B
     [,1] [,2] [,3] [,4]
[1,]    2   18    8   20
[2,]    6  -14   20    4
> A %*% B + t(A) %*% B
     [,1] [,2] [,3] [,4]
[1,]    2   18    8   20
[2,]    6  -14   20    4
```

```
> #(b)
> t(B) %*% B
     [,1] [,2] [,3] [,4]
[1,]    5   -6   17    8
[2,]   -6   65  -17   38
[3,]   17  -17   58   30
[4,]    8   38   30   52
> sum(diag(t(B) %*% B))
[1] 180
> B %*% t(B)
     [,1] [,2]
[1,]   90    9
[2,]    9   90
> sum(diag(B %*% t(B)))
[1] 180
> #(c)
> t(x) %*% t(B) %*% B %*% x
     [,1]
[1,] 1172
> t(B) %*% B %*% x %*% t(x)
     [,1] [,2] [,3] [,4]
[1,]   64  192  320  -64
[2,]   66  198  330  -66
[3,]  226  678 1130 -226
[4,]  220  660 1100 -220
> sum(diag(t(B) %*% B %*% x %*% t(x)))
[1] 1172
```

2. (i,j)원소가 $1/(i+j-1)$인 $n \times n$ 행렬 H를 힐버트(Hilbert) 행렬이라고 한다. $n = 2, \ldots, 10$까지 solve 함수를 이용하여 힐버트 행렬의 역행렬을 구해보고 역행렬이 존재하는지 답하시오.

- n = 2, 4, 6, 8, 10에 대하여, det함수를 통해서 구한 행렬식이 0이 아니고, 또한 Solve함수를 이용해서 구한 결과 모두 역행렬이 존재한다.

(1)

$$\begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}$$

```r
#Q2
#힐버트 행렬을 만들어주는 함수
make.Hibert = function(n)
{
  A = matrix(nrow =  n, ncol =  n)
  for(i in 1:n){
    for(j in 1:n){
      A[i,j] = 1/(i+j-1)
    }
  }
  return(A)
}
```

```
> make.Hibert(2)
     [,1]      [,2]
[1,]  1.0 0.5000000
[2,]  0.5 0.3333333
> det(make.Hibert(2))
[1] 0.08333333
> solve(make.Hibert(2))
     [,1] [,2]
[1,]    4   -6
[2,]   -6   12
> make.Hibert(4)
          [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.5000000 0.3333333 0.2500000
[2,] 0.5000000 0.3333333 0.2500000 0.2000000
[3,] 0.3333333 0.2500000 0.2000000 0.1666667
[4,] 0.2500000 0.2000000 0.1666667 0.1428571
> det(make.Hibert(4))
[1] 1.653439e-07
> solve(make.Hibert(4))
     [,1]  [,2]  [,3]  [,4]
[1,]   16  -120   240  -140
[2,] -120  1200 -2700  1680
[3,]  240 -2700  6480 -4200
[4,] -140  1680 -4200  2800
```

```
> make.Hibert(6)
          [,1]      [,2]      [,3]      [,4]      [,5]       [,6]
[1,] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000 0.16666667
[2,] 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667 0.14285714
[3,] 0.3333333 0.2500000 0.2000000 0.1666667 0.1428571 0.12500000
[4,] 0.2500000 0.2000000 0.1666667 0.1428571 0.1250000 0.11111111
[5,] 0.2000000 0.1666667 0.1428571 0.1250000 0.1111111 0.10000000
[6,] 0.1666667 0.1428571 0.1250000 0.1111111 0.1000000 0.09090909
> det(make.Hibert(6))
[1] 5.3673e-18
> solve(make.Hibert(6))
      [,1]    [,2]     [,3]     [,4]     [,5]     [,6]
[1,]    36    -630     3360    -7560     7560    -2772
[2,]  -630   14700   -88200   211680  -220500    83160
[3,]  3360  -88200   564480 -1411200  1512000  -582120
[4,] -7560  211680 -1411200  3628800 -3969000  1552320
[5,]  7560 -220500  1512000 -3969000  4410000 -1746360
[6,] -2772   83160  -582120  1552320 -1746360   698544
```

```
> make.Hibert(8)
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
[1,] 1.0000000 0.5000000 0.3333333 0.25000000 0.20000000 0.16666667 0.14285714 0.12500000
[2,] 0.5000000 0.3333333 0.2500000 0.20000000 0.16666667 0.14285714 0.12500000 0.11111111
[3,] 0.3333333 0.2500000 0.2000000 0.16666667 0.14285714 0.12500000 0.11111111 0.10000000
[4,] 0.2500000 0.2000000 0.1666667 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909
[5,] 0.2000000 0.1666667 0.1428571 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333
[6,] 0.1666667 0.1428571 0.1250000 0.11111111 0.10000000 0.09090909 0.08333333 0.07692308
[7,] 0.1428571 0.1250000 0.1111111 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857
[8,] 0.1250000 0.1111111 0.1000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667
> det(make.Hibert(8))
[1] 2.73705e-33
> solve(make.Hibert(8))
       [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
[1,]     64     -2016     20160     -92400    221760    -288288    192192    -51480
[2,]  -2016     84672    -952560    4656960  -11642400   15567552  -10594584   2882880
[3,]  20160   -952560   11430720  -58212000  149688000 -204324119  141261119 -38918880
[4,] -92400   4656960  -58212000  304919999 -800414996 1109908794 -776936155 216215998
[5,] 221760 -11642400 149688000 -800414996 2134439987 -2996753738 2118916783 -594593995
[6,] -288288 15567552 -204324119 1109908793 -2996753738 4249941661 -3030050996 856215352
[7,] 192192 -10594584 141261119 -776936154 2118916782 -3030050996 2175421226 -618377753
[8,] -51480   2882880  -38918880  216215998  -594593995  856215351  -618377753  176679358
> make.Hibert(10)
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]      [,10]
 [1,] 1.0000000 0.50000000 0.33333333 0.25000000 0.20000000 0.16666667 0.14285714 0.12500000 0.11111111 0.10000000
 [2,] 0.5000000 0.33333333 0.25000000 0.20000000 0.16666667 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909
 [3,] 0.3333333 0.25000000 0.20000000 0.16666667 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333
 [4,] 0.2500000 0.20000000 0.16666667 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333 0.07692308
 [5,] 0.2000000 0.16666667 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857
 [6,] 0.1666667 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667
 [7,] 0.1428571 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667 0.06250000
 [8,] 0.1250000 0.11111111 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667 0.06250000 0.05882353
 [9,] 0.1111111 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667 0.06250000 0.05882353 0.05555556
[10,] 0.1000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667 0.06250000 0.05882353 0.05555556 0.05263158
> det(make.Hibert(10))
[1] 2.164405e-53
> solve(make.Hibert(10))
             [,1]          [,2]          [,3]          [,4]          [,5]          [,6]          [,7]          [,8]          [,9]          [,10]
 [1,]  9.999719e+01 -4.949757e+03  7.919482e+04 -6.005529e+05  2522295.0 -6.305682e+06  9.608586e+06 -8.750620e+06  4.375286e+06 -9.236669e+05
 [2,] -4.949756e+03  3.266790e+05 -5.880152e+06  4.756344e+07  -208088462  5.350812e+08 -8.323439e+08  7.700561e+08 -3.898393e+08  8.313042e+07
 [3,]  7.919480e+04 -5.880151e+06  1.128980e+08 -9.512635e+08  4280662450 -1.123669e+10  1.775667e+10 -1.663324e+10  8.505608e+09 -1.828876e+09
 [4,] -6.005527e+05  4.756343e+07 -9.512634e+08  8.244246e+09 -37871868827  1.009913e+11 -1.615857e+11  1.528915e+11 -7.883455e+10  1.706956e+10
 [5,]  2.522294e+06 -2.080884e+08  4.280662e+09 -3.787187e+10  176734991839 -4.771836e+11  7.712047e+11 -7.357912e+11  3.820450e+11 -8.321430e+10
 [6,] -6.305679e+06  5.350810e+08 -1.123668e+10  1.009913e+11 -477183582308  1.301409e+12 -2.120813e+12  2.037577e+12 -1.064270e+12  2.330005e+11
 [7,]  9.608580e+06 -8.323436e+08  1.775667e+10 -1.615857e+11  771204559101 -2.120813e+12  3.480308e+12 -3.363622e+12  1.765901e+12 -3.883348e+11
 [8,] -8.750614e+06  7.700557e+08 -1.663323e+10  1.528915e+11 -735791094422  2.037577e+12 -3.363622e+12  3.267520e+12 -1.723107e+12  3.804102e+11
 [9,]  4.375282e+06 -3.898391e+08  8.505604e+09 -7.883452e+10  382044919568 -1.064270e+12  1.765901e+12 -1.723107e+12  9.122340e+11 -2.020931e+11
[10,] -9.236661e+05  8.313037e+07 -1.828875e+09  1.706955e+10  -83214282335  2.330005e+11 -3.883348e+11  3.804101e+11 -2.020931e+11  4.490964e+10
```

3. $(x_1, x_2, \ldots, x_6)^T = (10, 11, \ldots, 15)^T$라 하자. $(f(x_1), f(x_2), \ldots, f(x_6))^T = (25, 16, 26, 19, 21, 20)^T$을 만족하는 5차다항식 $f(x) = a_0 + a_1 x + \cdots + a_5 x^5$의 계수 $a_0, a_1, \ldots, a_5$의 값을 구하시오.

- 미지수 x와 y의 값들을 주어진 5차다항식에 대입하면, 아래와 같은 연립방정식을 구할 수 있습니다.
- 역행렬을 구하여 구한 해와 LU분해 강의 예제 코드를 통하여 구한 해가 같습니다.

$$\begin{pmatrix} 1 & 10 & 10^2 & 10^3 & 10^4 & 10^5 \\ 1 & 11 & 11^2 & 11^3 & 11^4 & 11^5 \\ 1 & 12 & 12^2 & 12^3 & 12^4 & 12^5 \\ 1 & 13 & 13^2 & 13^3 & 13^4 & 13^5 \\ 1 & 14 & 14^2 & 14^3 & 14^4 & 14^5 \\ 1 & 15 & 15^2 & 15^3 & 15^4 & 15^5 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} 25 \\ 16 \\ 26 \\ 19 \\ 21 \\ 20 \end{pmatrix}$$

```
> #Q3
> A = matrix(nrow =  6, ncol =  6)
> for(i in 1:6){
+   for(j in 1:6){
+     A[i,j] = (10 + i - 1)^(j-1)
+   }
+ }
> A
     [,1] [,2] [,3] [,4]  [,5]   [,6]
[1,]    1   10  100 1000 10000 100000
[2,]    1   11  121 1331 14641 161051
[3,]    1   12  144 1728 20736 248832
[4,]    1   13  169 2197 28561 371293
[5,]    1   14  196 2744 38416 537824
[6,]    1   15  225 3375 50625 759375
> B = c(25,16,26,19,21,20)
> #역행렬 구하기
> solve(A) %*% B
              [,1]
[1,]  2.536100e+05
[2,] -1.025510e+05
[3,]  1.650092e+04
[4,] -1.320667e+03
[5,]  5.258333e+01
[6,] -8.333333e-01
```

```
> #LU 분해 - 강의 예제 코드
> lufactorization = function(A){
+   n = nrow(A)
+   L = matrix(0,nrow=n,ncol=n)
+   for (k in (1:(n-1))){
+     for (i in ((k+1):n)){
+       L[i,k] = A[i,k]/A[k,k]
+       A[i,k] = 0
+       for (j in ((k+1):n)){
+         A[i,j] = A[i,j] - L[i,k]*A[k,j]
+       }
+     }
+   }
+   for (k in (1:n)){
+     L[k,k] = 1
+   }
+   return(cbind(L,A))
+ }
> lufactorization(A)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]  [,12]
[1,]    1    0    0    0    0    0    1   10  100  1000 10000 100000
[2,]    1    1    0    0    0    0    0    1   21   331  4641  61051
[3,]    1    2    1    0    0    0    0    0    2    66  1454  26730
[4,]    1    3    3    1    0    0    0    0    0     6   276   7950
[5,]    1    4    6    4    1    0    0    0    0     0    24   1440
[6,]    1    5   10   10    5    1    0    0    0     0     0    120
> aa = solve(lufactorization(A)[,1:6]) %*% B
> solve(lufactorization(A)[,7:12]) %*% aa
              [,1]
[1,]  2.536100e+05
[2,] -1.025510e+05
[3,]  1.650092e+04
[4,] -1.320667e+03
[5,]  5.258333e+01
[6,] -8.333333e-01
```

4. $n \times n$행렬 A에 대하여 $Ax = b$의 해를 구하는 partial pivoting을 적용한 가우스 소거법 프로그램을 작성하시오. 단, 입력값은 $n \times (n+1)$ 행렬인 $[A|b]$이고 출력값은 입력행렬에 소거법을 적용한 결과로 얻은 $n \times (n+1)$ 행렬이다. 작성한 프로그램을 다음의 방정식에 대하여 테스트 해보시오.

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 2 \\ -4 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

- Partial Pivoting을 이용해서 구한 방정식의 해와 원래의 행렬에서 역행렬을 구하는 Solve함수를 이용해서 구한 해가 같음을 알 수 있습니다.

```
67 gaussianeliminationpartial = function(Ab){
68   n = nrow(Ab)
69   for (k in (1:(n-1))){
70     pivotindex = k
71     for (i in ((k+1):n)){
72       if (abs(Ab[i,k]) > abs(Ab[pivotindex,k])){
73         pivotindex = i
74       }
75     }
76     if (pivotindex != k){
77       for (j in (k:(n+1))){
78         buffer = Ab[k,j]
79         Ab[k,j] = Ab[pivotindex,j]
80         Ab[pivotindex,j] = buffer
81       }
82     }
83     for (i in ((k+1):n)){
84       mik = Ab[i,k]/Ab[k,k]
85       Ab[i,k] = 0
86       for (j in ((k+1):(n+1))){
87         Ab[i,j] = Ab[i,j] - mik*Ab[k,j]
88       }
89     }
90   }
91   return(Ab)
92 }
```

```
> X = matrix(c(1,2,-4,2,4,2,3,2,1), nrow = 3)
> Y = c(1,2,3)
> X
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    4    2
[3,]   -4    2    1
> Y
[1] 1 2 3
> gaussianeliminationpartial(cbind(X, Y))
                 Y
[1,] -4 2 1.0 3.0
[2,]  0 5 2.5 3.5
[3,]  0 0 2.0 0.0
> gauss.X = gaussianeliminationpartial(cbind(X, Y))[,1:3]
> gauss.Y = gaussianeliminationpartial(cbind(X, Y))[,4]
> solve(gauss.X) %*% gauss.Y
 [,1]
-0.4
 0.7
 0.0
> solve(X) %*% Y
     [,1]
[1,] -0.4
[2,]  0.7
[3,]  0.0
```

5. 회귀모형 $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$에서 X와 $\boldsymbol{y}$가 다음과 같이 주어졌다.

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 3 \\ 1 & 5 & 4 \\ 1 & 5 & 4 \\ 1 & 7 & 5 \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} 2 \\ 4 \\ 5 \\ 8 \\ 8 \\ 9 \end{pmatrix}$$

(a) $X^T X$를 구하고, Choleski 분해에 의해 $X^T X = LL^T$로 분해되는 하삼각행렬 L을 강의 예제함수와 R의 내장함수를 이용하여 구하고 비교하시오.

(b) X에 대한 QR분해를 이용하여 회귀계수의 추정값 $\hat{\beta}$를 구하고 lm 함수에 의하여 구한 회귀계수의 추정값과 비교하시오.

```
#강의 예제 함수
choleskyfactorization = function(A){
  n = nrow(A)
  L = matrix(0,nrow=n,ncol=n)
  for (i in (1:n)){
    L[i,i] = A[i,i]
    if (i > 1){
      for (k in (1:(i-1))){
        L[i,i] = L[i,i] - L[i,k]*L[i,k]
      }
    }
    L[i,i] = (L[i,i])^(1/2)
    if (i < n){
      for (j in ((i+1):n)){
        L[j,i] = A[j,i]
        if (i > 1){
          for (k in (1:(i-1))){
            L[j,i] = L[j,i] - L[j,k]*L[i,k]
          }
        }
        L[j,i] = L[j,i]/L[i,i]
      }
    }
  }
  return(L)
}
```

```
> #Q5 - (a)
> # Choleski decomposition
> X = matrix(c(1,1,1,1,1,1,2,3,5,5,7,1,3,3,4,4,5), nrow = 6)
> Y = c(2,4,5,8,8,9)
> X
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    2    3
[3,]    1    3    3
[4,]    1    5    4
[5,]    1    5    4
[6,]    1    7    5
> Y
[1] 2 4 5 8 8 9
> choleskyfactorization(t(X) %*% X)
         [,1]     [,2]     [,3]
[1,] 2.449490 0.000000 0.000000
[2,] 9.389711 4.983305 0.000000
[3,] 8.164966 2.876270 1.029759
> t(choleskyfactorization(t(X) %*% X))
         [,1]     [,2]     [,3]
[1,] 2.44949 9.389711 8.164966
[2,] 0.00000 4.983305 2.876270
[3,] 0.00000 0.000000 1.029759
> chol(t(X) %*% X)
         [,1]     [,2]     [,3]
[1,] 2.44949 9.389711 8.164966
[2,] 0.00000 4.983305 2.876270
[3,] 0.00000 0.000000 1.029759
```

(a) - 강의 예제 함수를 적용했을 경우 하삼각행렬이 나오는데, 이 행렬을 Transpose 해서 상삼각행렬로 만들어 주면 R의 내장 함수인 chol함수의 결과값과 동일하다.

(b) - 주어진 행렬 X 가 X = QR 로 분해 가능하다고 할 때, $X\beta = Y = QR\beta$ 이므로

$R\beta = Q^T Y$로 변환하여 회귀계수를 구할 수 있다.

- QR분해, qr.solve() 함수, lm 함수를 이용해서 구한 회귀계수의 값은 모두 같음을 알 수 있습니다.

```
> #Q5 - (b) : Regression with QR Decomposition
> X = matrix(c(1,1,1,1,1,1,2,3,5,5,7,1,3,3,4,4,5), nrow = 6)
> Y = c(2,4,5,8,8,9)
> X
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    2    3
[3,]    1    3    3
[4,]    1    5    4
[5,]    1    5    4
[6,]    1    7    5
> Y
[1] 2 4 5 8 8 9
> X.qr = qr(X)
> qr.Q(X.qr)
           [,1]       [,2]         [,3]
[1,] -0.4082483 -0.5685651  0.677815759
[2,] -0.4082483 -0.3678950 -0.703885596
[3,] -0.4082483 -0.1672250 -0.143384103
[4,] -0.4082483  0.2341150  0.006517459
[5,] -0.4082483  0.2341150  0.006517459
[6,] -0.4082483  0.6354551  0.156419021
```

```
> qr.R(X.qr)
         [,1]      [,2]      [,3]
[1,] -2.44949 -9.389711 -8.164966
[2,]  0.00000  4.983305  2.876270
[3,]  0.00000  0.000000 -1.029759
> solve(qr.R(X.qr)) %*% t(qr.Q(X.qr)) %*% Y
          [,1]
[1,] 0.6455696
[2,] 0.8354430
[3,] 0.6455696
> qr.solve(X, Y)
[1] 0.6455696 0.8354430 0.6455696
> lm(Y ~ X)$coef
(Intercept)          X1          X2          X3
  0.6455696          NA   0.8354430   0.6455696
```

6. $A = \begin{pmatrix} 2 & -4 & 2 \\ 4 & -9 & 7 \\ 2 & 1 & 3 \end{pmatrix}$에 대한 SVD는 $A = UDV^T$라 하자. A^TA의 고유값을 D의 대각원소의 값과 비교하시오. 둘 사이의 관계는 무엇인지 답하시오.

$A^TA = VD^TU^TUDV^T = VD^TDV^T$가 성립하므로 A^TA의 고유값은 D의 대각원소의 제곱과 같다.

또한, 아래의 사진에서 V 행렬의 각 열은 A^TA의 고유벡터이다.

```
> #Q6
> A = matrix(c(2,-4,2,4,-9,7,2,1,3), nrow = 3, byrow = T)
> A
     [,1] [,2] [,3]
[1,]    2   -4    2
[2,]    4   -9    7
[3,]    2    1    3
> t(A) %*% A
     [,1] [,2] [,3]
[1,]   24  -42   38
[2,]  -42   98  -68
[3,]   38  -68   62
> eigen(t(A) %*% A, symmetric = T)
$values
[1] 171.912180  11.573134   0.514686

$vectors
           [,1]      [,2]         [,3]
[1,] -0.3568640 0.3871518  0.850153820
[2,]  0.7341837 0.6789501 -0.001003374
[3,] -0.5776005 0.6238110 -0.526533452

> svd(A)
$d
[1] 13.1115285  3.4019309  0.7174162

$u
           [,1]        [,2]       [,3]
[1,] -0.3665220 -0.20396492 -0.9077775
[2,] -0.9211979 -0.05739861  0.3848373
[3,] -0.1305985  0.97729407 -0.1668542

$v
           [,1]      [,2]         [,3]
[1,] -0.3568640 0.3871518 -0.850153820
[2,]  0.7341837 0.6789501  0.001003374
[3,] -0.5776005 0.6238110  0.526533452

> (svd(A)$d)^2
[1] 171.912180  11.573134   0.514686
```