```
> # This sciprt is also stored in
> #
> #      matrix.r
> #----------------------------
> #   Inverse of a matrix
> #----------------------------
>
>  w<-matrix(c(1,2,3,4,5,6,7,8,10),3,3,byrow=T)
>  w
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
>
>  winv<-solve(w)
>  winv
           [,1]      [,2] [,3]
[1,] -0.6666667 -1.333333    1
[2,] -0.6666667  3.666667   -2
[3,]  1.0000000 -2.000000    1
>
>  w%*%winv
             [,1]          [,2]          [,3]
[1,] 1.000000e+00 -4.440892e-16 -1.110223e-16
[2,] 4.440892e-16  1.000000e+00 -2.220446e-16
[3,] 4.440892e-16  8.881784e-16  1.000000e+00
>
> # ------------------------------
> #   Determinant of a matrix
> # ------------------------------
>
>
> # Build your own function
>
>  determ<-function(M) Re(prod(eigen(M, only.values=T)$values))
>
>  determ(w)
[1] -3
>
> #  Another function
>
>  absdet <- function(M) abs(prod(diag(qr(M)$qr)))
>
>  absdet(w)
[1] 3
>
```

```
> #  Another example
>
>  x1 <- matrix(c(1,2,3,4,5,6,7,8,9),ncol=3,byrow=T)
>  x1
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
>
>  determ(x1)
[1] 2.199534e-14
>
>  absdet(x1)
[1] 5.350789e-15
>
>
> #--------------------------------------------
> #  Rank of a matrix:  use the "qr" function
> #--------------------------------------------
>
>
>   A <- matrix(c(1, 1,  1,
+               2, 5, -1,
+               0, 1, -1),3,3,byrow=T)
>
>  A
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    5   -1
[3,]    0    1   -1
>
>   qr(A)$rank
[1] 2
>
>
> #   Another example
>
>   A <- matrix(c(1,1, 1,
+               2,5,-1,
+               0,1, 1),3,3,byrow=T)
>
>   A
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    5   -1
[3,]    0    1    1
>
>   qr(A)$rank
[1] 3
>
```

```
> #  Another example
>
>   X <- matrix(c(1,1,0,0,
+               1,1,0,0,
+               1,0,1,0,
+               1,0,1,0,
+               1,0,0,1,
+               1,0,0,1),ncol=4,byrow=T)
>    X
     [,1] [,2] [,3] [,4]
[1,]    1    1    0    0
[2,]    1    1    0    0
[3,]    1    0    1    0
[4,]    1    0    1    0
[5,]    1    0    0    1
[6,]    1    0    0    1
>
>    qr(X)$rank
[1] 3
>
>
> # Note that the sum of squares
> # and crossproducts matrix has
> # the same rank as X
>
>  XtX <- t(X)%*%X
>  XtX
     [,1] [,2] [,3] [,4]
[1,]    6    2    2    2
[2,]    2    2    0    0
[3,]    2    0    2    0
[4,]    2    0    0    2
>
>  qr(XtX)$rank
[1] 3
>
>
> # This is a square symmetric matrix
> # but the inverse does not exist
>
>  solve(XtX)
이하에 에러solve.default(XtX) :
  시스템은 수치적으로 특이합니다: 조건수의 역수  = 1.38778e-17
>
>
> #  Note that the function "rank" R
> #  is related to sorting.  It computes the
> #  ranks of the elements of a vector.
>
```

```
>    rank(c(1.2, 5.1, 3.5, 9.8))
[1] 1 3 2 4
>
> # ---------------------------------
> #  Create an nxn identity matrix
> # ---------------------------------
>
>
>    diag(rep(1,4))
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
>
>
> #--------------------------
> #  Trace of a matrix
> #--------------------------
>
>  w<-matrix(c(1,2,3,4,5,6,7,8,10),3,3,byrow=T)
>  w
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
>
>  tr<-sum(diag(w))
>  tr
[1] 16
>
> #---------------------------------
> #  Compute row sums or column sums
> #---------------------------------
>
>  sum(w)
[1] 46
>
>  apply(w,1,sum)
[1]  6 15 25
>
>  apply(w,2,sum)
[1] 12 15 19
>
>  apply(w,1,prod)
[1]   6 120 560
>
>  apply(w,1,mean)
[1] 2.000000 5.000000 8.333333
>
```

```
>  apply(w,1,var)
[1] 1.000000 1.000000 2.333333
>
```