```
# Data structure: 2. Matrix

# Matrix is a two-dimensional array of numbers or strings.
mymat <- matrix(data=1:30, nrow=3, ncol=10)   # mymat <- matrix(1:30, 3, 10)
mymat
mymat1 <- matrix(1:30, 3, 10, byrow=TRUE)     # Fill each row first.
mymat1
mymat1_1 <- matrix(mymat1,30,1)

is.matrix(mymat)
class(mymat)

mymat2 <- matrix(state.abb,5,10)
dim(mymat2)
nrow(mymat2)
ncol(mymat2)
length(mymat2)

rownames(mymat) <- LETTERS[1:3]
colnames(mymat) <- letters[1:10]
mymat

# submatrix seleciton: row and column numbers, names, logical vectors.
mymat[c(1,3),]
mymat[2:3,c(3,5)]
mymat["A",]
mymat[c("A","C"),letters[2:5]]
mymat[,colnames(mymat) > "d"]
mymat[c(TRUE,FALSE,TRUE),]

col(mymat1)
row(mymat1)
mymat_up <- col(mymat1) >= (row(mymat1) + 3)
mymat_up
mymat1*mymat_up                           # entrywise multiplication

# Matrix operations
trans_mymat <- t(mymat)
ele_pro <- mymat1 * mymat                 # entrywise multiplication
mat_pro <- t(mymat1) %*% mymat            # matrix multiplication
mat_pro2 <- crossprod(mymat1, mymat)      # since the second argument is the same,
mymat / mymat1                            # entrywise

sum(mat_pro)                              # sum of all the elements
colSums(mymat)                            # colMeans(mymat), rowMeans(mymat)
rowSums(mymat)

# inverse matrix

A <- matrix(rnorm(25), 5, 5)
Ainv <- solve(A)                          # inverse matrix
b <- 1:5
x <- solve(A,b);                          # solve the system of equations b = Ax
solve(A)%*%b
b_mat <- as.matrix(b)

# Combine matrices
cbind(mymat,mymat1)
rbind(mymat,mymat1)

# Exercise 1 #####################################################################

# Construct the following two matrices using

a <- 1:9

#  1  4  9              1 16 49
# 16 25 36     and      4 25 64
# 49 64 81              9 36 81
#  1  4  9  upper triangular matrix
#  0 25 36
#  0  0 81


# Exercise 2 ######################################################################

# Housing price in CT
Storrs <- c(365, 489)
Hartford <- c(426, 387)
Stamford <- c(571, 486)

# Create HP_vector
# Using c(.) to combine the three vectors into one vector HP_vector

HP_vector <-

# Construct HP_matrix

# Construct a matrix with 3 rows, where each row represents an area.
# Use the matrix() function, with nrow =3, byrow=TRUE
HP_matrix <-

# Name the matrix: Define the column names (House and Condo)
# and row names (Storrs, Hartford, Stamford)

Area <- c("Storrs","Hartford","Stamford")
Type <-c("House","Condo")



# Calculate the average prices of house and condo.
CT_average_type <-


# Combine CT_average with HP_matrix. Include CT_average in HP_matrix at the fourth row.
# Assign this new matrix to a variable HP_matrix.2
HP_matrix.2 <-

# Caculate the average of House and Condo price in each row of HP_matrix.2
# Assign this new matrix to a variable HP_matrix.f
CT_average_area <-
HP_matrix.f <-

# Exercise 3 ######################################################################

# 1. Generate a vector of 100 random numbers from chi-square distribution with df=5 and
#    assign this vector to Income.
income <-

# 2. Randomly choose 100 numbers from {7,...,16} with replacement and assign this vector
#    to yrsofedu. You can use sample() function.
yrsofedu <-

# 3. Construct a (100 x 2) matrix, CT, in which the first column is Income and the second is
#    Income.
CT <- cbind(income,yrsofedu)
# 4. Randomly choose between "Male" and "Female" 100 times and assign this character
#    vector to gender. We can think each element of this vector represent the gender
#    corresponding to the same row of CT.
gender <-

# 5  Create CT1, the submatrix of CT, which contains only Females' income and years of education
CT1 <-

# 6. Create CT2, the submatrix of CT, which contains income and years of education
# for higher education people (years of education > 12)
CT2 <-

# 7. Calculate the average incomes in CT1 and CT2.

colMeans(CT)


# Exercise 4: Run multivariate OLS####################################################

x1 <- rnorm(100,2,4)
x2 <- rchisq(length(x1),3)
e <- rnorm(length(x1),0,4)

# Generate Y based on Y = 2 + 3*x1 + 2*x2 + e
Y <- 4 + 3*x1 + 2*x2 + e
# Construct a matrix X that contains the constant term and two regressors.
X <-
# X <- cbind(1,x1,x2)
# Using the multivariate OLS formula and matrix operations, obtain the estimates of
# coefficients and assign them to a vector beta_hat.

b_hat <-
lm(Y~x1+x2)


```