```
# Important data structures: include vectors, matrices, data frames and lists.

# 1. vector: sequence of values, all of the same type (numeric, logical, character)

# Numeric vectors

# The simplest data type is the numeric vector. Each element of this vector is a number.
# To set up a vector, we use the function c() as follows:

weight <- c(60, 72, 57, 90)
is.vector(weight)
is.numeric(weight)
length(weight)

name <- c("Graham", "James", "Victor", "Ivan")
names(weight) <- name

mu <- mean(weight)
sigma1 <- sqrt(sum((weight - mu)^2)/(length(weight)-1))
sigma <- sd(weight)
total <- sum(weight)
med <- median(weight)

# Operation with vectors is carried out elementwise.
height <- c(1.75, 1.80, 1.65, 1.90)
bmi <- weight/height^2
bmi

1:3
1:3 + 1:3
1:3 * 1:3               # Elementwise
1:3*2
(1:3)*2
1:3^2
(1:3)^2
seq(2,18,8)/1:3
abs(-2:2)
sqrt(c(81,100,121))
x <- exp(1:3)
log(x)

# Recycling rule: Shorter vectors are recycled as often as needed until they match
# the length of the longest vector.
x <- c(10, 5, 3)
y <- c(x, x, 1)
v <- 2*x + y + 1

# how to create vectors
x1 <- seq(4,9)
x2 <- seq(4,10,2)
x3 <- 4:9
x4 <- rep(x1,3)
x5 <- rep(x2,1:4)
x6 <- c(x3,NA)                         # Not available/Missing values

mean(x6)                               # Does not work because of NA
mean(x6,na.rm=TRUE)                    # Ignore NA to take the mean

# Index vector, selecting subsets of a data set

set.seed(pi)                           # The seed number you choose is the starting
point
                                       # used to generate a sequence of
                                       # random numbers. You can obtain the same numbers
                                       # with the same seed number.

z <- runif(20, min = -1, max = 1)
z1 <- z[1:10]
z2 <- z[-(1:10)]
```

```
print(min(z))
print(which.min(z))        # where is the minimum in z?
print(which(z>0))          # where is z>0?

x <- rep(NA,6)
x[2*(1:3)] <- (1:3)^2
x[2*(1:3)-1] <- -(1:3)^2

# Floor, ceiling and round functions
x <- c(1.3411,4.603,3.5)
x_round1 <- round(x)
x_round2 <- round(x,2)
x_floor <- floor(x)
x_ceiling <- ceiling(x)

# Exercises  ######################################################################

# Using the vector operation in R (not lm function!), obtain the slope cofficient.
set.seed(2)
x <- rchisq(20,5)                          # Generate a (20X1) vector of random
                                           # numbers from chi-square distribution
                                           # with df=5
e <- rnorm(length(x),0,2)                  # Generate a vector of normal random numbers
                                           # with mean=0 and sd=5. The dimension is
                                           # the same as the vector x
y <- 3 + 2*x + e                           # Generate y = 3 + 2*x + e

# Using runif and round functions or sample function, generate 100 random draws from
# (0,1) and (-1,0,1) and print the indices of these two vectors that have the same values.

a <- runif(100)
a <- round(a)
b <-
b <- round(b)

same <-                                    # or you can use which(a==b)
b[same]

# You played poker and roulette and collected your winnings and losses
#  ------------------------------------------------------------------
#             Mon     Tue     Wed     Thu     Fri
#  poker     $140    -$50     $20   -$120    $240
#  roulette  -$24    -$50    $100    $350     $10
#  ------------------------------------------------------------------

# Create two vectors: poker_vector and roulette_vector
poker_vector <- c(140, -50, 20, -120, 240)
roulette_vector <-

# Assign days as names of poker_vectors and roulette_vector
days_vector <- c("Mon","Tues","Wed","Thu","Fri")
names(poker_vector) <-
names(roulette_vector) <-

# From poker_vector, select the poker result from Tuesday and Thursday and assign to
# poker_midweek
poker_midweek <-
# Calculate the average earning in poker_midweek
ave_poker_midweek <-

# What is the biggest amount you made in poker? When did you make it?
print(max(poker_vector))
print(which.max(poker_vector))
print(which(poker_vector > roulette_vector))

# Money you made in poker and in roulette
total <- poker_vector + roulette_vector

```

```
# Calculating total weekly winnings with poker
total_poker <- sum(poker_vector)
total_roulette <-

# Comparing total winnings
total_poker > total_roulette


```