```r
# Logical Vectors

# Logical vectors can take the values, "TRUE" and "FALSE".

# Using the logical vector, you can pick elements of a vector that satisfy some conditions.
# Alternatively, you can use "which()" function.

# Index vector, selecting subsets of a data set
a <- c("STORRS","HARTFORD","STAMFORD")
a[c(TRUE,FALSE,TRUE)]
a[which(a > "S")]

z <- runif(10, min = -1, max = 1)
z
plus <- z>0
z_plus <- z[plus]
sum(plus)                      # Count the number of positive numbers in z

ind_plus <- which(z>0)
z[ind_plus]

z_minus <- z[!(z>=0)]          # ! represents "not"
z[z<0] <- -z[z<0]

# Exercise #############################################################################################

a <- rnorm(20,6,2)

# From the vector a,
# 1. Generate a vector a_mid that contains elements that is between 4 and 7 (including 4 and 7).
# Which one is correct?

# 2. Also present the indices of a that belongs to this. (hint: which)

# 3. Generate a vector a_end that contains elements that are greater than 9 or less than 3.


# Generate a fake income dataset of size 1000 from based on chi square distribution with degree of
# freedom 8
income <- rchisq(1000, 8)
# Find the range of middle income that belong to 40% - 60%, using "quantile()" function.
?quantile

# Identify the indices of individuals in this group.


########################################################################################################

# TRUE and FALSE are also recognized as 1 and 0 respectively
b <- c(TRUE, TRUE, FALSE, TRUE)
is.logical(b)
is.logical(b) + is.numeric(b)
b1 <- as.numeric(b)                       # as.numeric() is the function to transfer to the numeric vector.
class(b1)

z <- runif(10, min = -1, max = 1)
plus <- z>0
sum(plus)                      # Count the number of positive numbers in z

# You can also try this.
vec <- c(3, FALSE, 5, 6, 2, NA, pi);
```

```
class(vec);
vec[is.na(vec)] <- 0
vec2 <- c("a","b",TRUE)
vec2
class(vec2)

# Evaluate how one object is related with another object

c(TRUE,TRUE) & c(TRUE,FALSE)
c(TRUE,TRUE) && c(TRUE,FALSE)    # Count only the inital element of each vector.
c(TRUE,TRUE) && c(TRUE,FALSE)
c(TRUE,TRUE) && c(FALSE,TRUE)

c(TRUE,FALSE) | c(FALSE,FALSE)
c(TRUE,FALSE) || c(FALSE,FALSE)
c(FALSE,TRUE) || c(FALSE,FALSE)

4 %in% 1:5                       # belong to
4 %in% 6:10
"hi" %in% c("hola","hi","hello")
"hi" %in% c("hola","hide","hello")

# Comparison of logicals
TRUE == FALSE
# Comparison of numerics: Check if -6*14 is not equal to 17-101
-6*14 != 17-101
# Comparison of character strings: Are "useR" and "user" equal?
"useR" == "user"
# Compare a logical with a numeric
TRUE == 1

#Greater and less than
# Comparison of numerics : check if -6*5+2 is greater than or equal to -10+1
-6*5+2 >= -10+1
# Comparison of character strings: check "raining" is less than or euqal to "raining dogs"
"raining" <= "raining dogs"
# Comparison of logicals : Is TRUE greater than FALSE?
TRUE > FALSE

# Compare vectors
# The number of views
linkedin <- c(16,9,13,5,2,17,14)
facebook <- c(17,7,5,16,8,13,14)
week <- c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
names(linkedin) <- week
names(facebook) <- week

# popular days: when did the number of linkedin views exceed 15?
linkedin > 15
# Quiet days: which days was the linkedin views only 5 times or fewer
linkedin <= 5
# When was the linedin views more than facebook views
linkedin > facebook


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

# State all the state names whose initial letter is Between "F" and "U"? (hint: state.name)


#  -------------------------------------------------------------------
#             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 <- c(-24, -50, 100, 350, 10)

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



# Which days did you make money on poker and how much did you make on those days
selection_vector <-
poker_winning_days <-

# Select the days that you made money both in poker and in roulette
# Calculate how much you made on those days.
winning_days <-
total <- poker_vector + roulette_vector
total_winning_days <-

# Count the number of days that you won in poker or in roulette


# Generate a (1000 x 1) vector of standard normal random variables.
# See whether each element is > 1.96 or < -1.96.
# Count this number and calculate the proportioan of this number to 1000.
# Recall that we did the same exercise using for loop.
```