```r
###### COURSE: Basics of R programming language for statistical analysis #######

#Instructor: Marina FERENT-PIPAS
#email: marinaferent@gmail.com
#Multicultural Business Institute | mcb-institute.org

#https://tinyurl.com/recapQuizzMeet6
#https://tinyurl.com/cheatSheet00

#_______ CHAPTER 2: CONTROL STRUCTURES AND FUNCTIONS | Statistical measures _____#
########## MEETING 06: FUNCTIONS| Challenge: Mode values| 14.09.2021 #############

#_____________________________________________________________________________#
#################### GOOD PROGRAMMING PRACTICES OF THE MEETING:
    ##1. DESIGN: Before coding a section - write down each step in order
    ##2. INDENTATION: Showcase the begining of a section, the body and the end of it
    through the right indentation.
            ##Anything that subordinates to a line is TABed once from that line.

#################### FOCUS OF THE MEETING: FUNCTIONS
##structure:
###functionName=function(functionArguments)
###{
        #What the function does with the arguments
###}
#_____________________________________________________________________________#

#INTRO DEFINING A FUNCTION IN R#

#EXERCISE 1: Comment the code below:

suma=function(x,y)    #the name of the function is suma and it has 2 arguments - x and y
{                        #here begins what the function does
  x+y                 #suma returns the value of x+y
}                        #here ends what the function does

suma(5,6)            #append the function suma => adds 5+6 and returns 11

#EXERCISE 2: Define a function named patrat that will return the squared value of a
given value.
    #If I type in patrat(5) it should return 25

patrat=function(x)
{
    x^2
}
    ##alternatively:

patrat=function(x)
{
    x*x
}

patrat(5)

#EXERCISE 3: Comment the code below. What is the difference between function suma (in
EXERCISE 1 above) and function sumaVector below?

sumaVector=function(x) #the name of the function is sumaVector and it has 1 argument-x
{
  suma=0                 #initialize suma. giving it the value 0
  for(i in 1:length(x))
  {                      #we add up all the values in vector x in suma: suma=0+first
  value; suma=suma+second value etc.
    suma=suma+x[i]
  }
  suma                   #return suma
}

```

```r
#EXERCISE 4: Define a vector called vector having the values 1, 4 and 5. Add up all the values of vector vector using the function sumaVector defined above.
vector=c(1,4,5)
sumaVector(vector)

#EXERCISE 5: Define a function lungimeVector that will return:
    ###if length(x)<10 => "short vector"
    ###if length(x)>=10=>"long vector"
        #e.g: vector1=5:9=> lungimeVector(vector1) will return "short vector"
            # vector2=1:50=> lungimeVector(vector2) will return "long vector"

lungimeVector=function(x)
{
  if (length(x)<10){
    print("short vector")
  } else {
       print("long vector")
         }
}

vector1=5:9
vector2=1:50
lungimeVector(vector1)
lungimeVector(vector2)

#EXERCISE_POINT_1: Why is it that the following code works the same way? (no error)

lungimeVector=function(x)
  if (length(x)<10){
    print("short vector")
  } else {
       print("long vector")
         }


#__________ TODAY'S R CHALLENGE: Mode values _____________#

#EXERCISE 1: Import the <<Campus crimes.csv>> into R. Save it as campusData.
setwd("E:/Work/Multicultural Business Institute/R/R_FBE2021/Meeting 6") #sets the working directory. all the files exported/imported following this line, are exported to/imported from this location
campusData=read.csv("Campus crimes.csv")                ##imports the campusData from a .csv file called Campus crimes


#EXERCISE 2: Compute the mode for variable privateCollege in campusData.

    ##Step 1: Compute the table of frequencis
View(campusData) #=> I observe that variable privateCollege is on the second column
table(campusData[,2]) #=> I compute the absolute frequencies for variable privateCollege
    #alternatively: table(campusData$privateCollege)
absFreq=as.data.frame(table(campusData[,2])) #store the absolute frequencies table in absFreq data frame for future use
    ##Step 2: I check the absolute frequencies table:
View(absFreq)   #the states of the variable are in column 1 and the frequencies in column 2
    ##Step 3: I observe that the highest frequency is 85=> row 2; I conclude that the mode is the value in row 2, column 1
mode=absFreq[2,1]

        ##alternatively - I could ask R to find the row with the highest frequency
max(absFreq[,2]) #I search for the maximum value in the second column (the column with frequencies)
                  #it returns 85
mode=absFreq[absFreq[,2]==max(absFreq[,2]),1] #the mode will be the element in column 1, on the row that has the maximum value on column 2
```

```r
#EXERCISE 3: Write a function that computes the mode.

mode=function(x)
{
  absFreq=as.data.frame(table(x))
  modeValue=absFreq[absFreq[,2]==max(absFreq[,2]),1]
  modeValue
}

mode(campusData[,2])

#EXERCISE 4: Write a function that computes and interprets the mode of a variable in a dataframe.

mode=function(x)
{
  absFreq=as.data.frame(table(x))
  modeValue=absFreq[absFreq[,2]==max(absFreq[,2]),1]
  print(paste0("the mode is ", modeValue))
}

mode(campusData[,2])

#EXERCISE_POINT_2: Find the R defined function that computes the mode of a variable.
#EXERCISE_POINT_3: Write a function that computes and interprets the mode of all the variables in a dataframe.
#EXERCISE_POINT_4: Write a function that computes and interprets the skewness of a variable.
    #(Interpretation: Skew=0 => the distribution is symmetric; Skew<0 => the distribution has negative assymetry; Skew>0=> the distribution has positive assymetry)
#EXERCISE_POINT_5: Write a function that computes and interprets the kurtosis of a variable.
    #(Interpretation: Kurt=3 => normal distribution; Kurt<3=> platikurtic distribution; Kurt>3=> leptokurtic distribution)
#EXERCISE_POINT_6.1: Write a function that computes the mean of a variable.
#EXERCISE_POINT_6.2: Write a function that computes the means of multiple variables in a data set and stores them in a data frame.
#EXERCISE_POINT_7.1: Write a function that computes the median of a variable.
#EXERCISE_POINT_7.2: Write a function that computes the medians of multiple variables in a data set and stores them in a data frame.
#EXERCISE_POINT_8: Write a function that computes and interprets the coefficient of variation of a variable.
        #(Interpretation: CV>30% => the mean is not representative, the population is heterogenous or CV<30% => the mean is representative, the population is homogenous).
#EXERCISE_POINT_9: Comment EXERCISE 3 and 4 above.
```