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

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

#################### MEETING 02: Basic notions | 19.08.2021 ####################

#################### 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.
            ###e.g. I want to export a graph:
            ###png("age.png") - beggining of the section, tells R that I want to save
            ###[TAB 1] barplot(ageFreq, main="age distribution", xlab="age", ylab="no. participants") - the body, tells R which graph I want to save
            ###[NO TAB] - eding section, tells R I'm closing the saving function

#################################################################################
#@You are given the following information in regards to the participants to an R programming course:
age=c(20, 21, 23, 24, 22, 22) #the age of the participants
education=c("HS", "BA", "MSc", "MSc", "BA", "BA") #the education level of the participants
pass=c("PASSED", "FAILED", "PASSED", "PASSED", "PASSED", "PASSED") #whether they passed or failed the course

#_______________________________________
#DEFINE A DATA FRAME/MATRIX. PROPERTIES.
#*Numerical representation of attributive statistical variables: distributions in absolute and relative frequencies

#*EXERCISE 1: Represent the distribution of participants with respect to age in absolute frequencies.
absFreq=as.data.frame(table(age))   #the function table sorts the values of vector age and counts their frequency
View(absFreq)                       #visualize the data frame

dim(absFreq)    #returns the number of rows and the number of columns of a data frame/matrix
nrow(absFreq)   #returns the number of rows of a data frame/matrix
ncol(absFreq)   #returns the number of columns of a data frame/matrix
names(absFreq)  #returns the colnames of a data frame/matrix

absFreq[3,2]    #returns the element in row 3, col 2
absFreq[3,]     #returns elements in row 3
absFreq[,2]     #returns all the elements in col 2

absFreq$age     #returns all the values in column age =>vector age | nameDataFrame$colName (I know the name of the column from names(absFreq) above)
absFreq$Freq    #returns all the values in column Freq => vector Freq| nameDataFrame$colName (I know the name of the column from names(absFreq) above)


#######____Discussion point:
absFreq_tab=table(age)

is.data.frame(absFreq_tab) #the answer is false
is.table(absFreq_tab)      #the answer is true
#=>absFreq_tab is a table and not a data frame

is.data.frame(absFreq) #the answer is true
is.table(absFreq)      #the answer is false
#=>absFreq is a data frame and not a table
```

```
##What are the differences between a table of frequencies and a data frame?

    View(absFreq)
    View(absFreq_tab)

    dim(absFreq)
    dim(absFreq_tab)

    nrow(absFreq)
    nrow(absFreq_tab)

    ncol(absFreq)
    ncol(absFreq_tab)

    rownames(absFreq)
    rownames(absFreq_tab)

    colnames(absFreq)
    colnames(absFreq_tab)

    absFreq$age
    absFreq_tab$age

#######____END discussion point


#*EXERCISE 2: How many students are 22 years old? (ABSOLUTE FREQUENCIES)
View(absFreq)               #visualize the data frame
#I observe from above that age 22 is in row 3 and that the frequencies are in column 2 => my frequency is in row 3, column 2
absFreq[3,]                 #returns elements in row 3 => 22,2 the first element is the age, the second the frequency so I know the frequency is 2
    ###alternatively:
absFreq[3,2]                #returns the element in row 3, column 2 => 2 = the frequency I was searching for
    ###alternatively:
absFreq[absFreq$age==22,2]  #returns the element in row where age is 22 and in column 2


#*EXERCISE 3: What percent of students are 22 years old? (RELATIVE FREQUENCIES)
total=length(age)           #total number of students (=sample size)
absFreq_22=absFreq[absFreq$age==22,2]   #number of students aged 22
relFreq_22=absFreq_22/total*100         #percent of students aged 22
relFreq_22                              #returns the relative frequency without % sign
print(paste0(relFreq_22, "%"))          #returns the relative frequency % sign
print(paste0("the percent of students aged 22 is ", relFreq_22, "%")) #function paste0 combines text with vector value

    ###alternatively:
total=sum(absFreq[,2])          #sums up all the values in second column = adds up the frequencies = sample size = total number of students

relFreq=absFreq$Freq/total      #computes the vector of relative frequencies as the absolute freguencies divided to the total nuber of students
relFreq                         #returns the relative frequencies without the age
freq=cbind(absFreq,relFreq)     #combines/concatenates by columns the absFreq data frame with the relFreq vector
View(freq)
freq[absFreq$age==22,3]         #returns the element in row where age is 22 and in column 3 (col 3=rel frequency)
print(paste0(freq[absFreq$age==22,3]*100, "%")) #returns the relative frequency % sign

#EXERCISE_POINT_1: Change the column names of the absFreq data frame to <<"age" | "absFreq" | "relFreq">>. At this point the names are <<"age" | "Freq" | "V3">>.
#EXERCISE_POINT_2: Compute the absolute and relative frequencies for variable <<education>> and <<pass>>.
```

```r
#*EXERCISE 4: Combine the vectors "age", "education" and "pass" into a table such that the values of age are in column 1, education in column 2, and pass in column 3. Name your table participantsInfo.
participantsInfo=cbind(age,education, pass) #combines/concatenates by columns the vectors age, education and pass
View(participantsInfo)

###____Discussion point:
participantsInfo_rows=rbind(age,education, pass)  #combines/concatenates by rows the vectors age, education and pass
View(participantsInfo_rows)
###____END discussion point


#*EXERCISE 5: A 7th participant joins the course:
seven=c(23, "BA", "FAILED")
    #Combine "participantsInfo" with vector "seven" into "participantsInfoNew" such that "seven" is the las row of "participantsInfoNew".
participantsInfoNew=rbind(participantsInfo, seven) #combines/concatenates by row participantsInfo and vector seven
View(participantsInfoNew)


#*EXERCISE 6: Save "participantsInfoNew" into <<Participants Information.csv>> file.
write.csv(participantsInfoNew, "E:/Work/Multicultural Business Institute/R/R_FBE2021/Meeting 2/Participants Information.csv")
    #exports the participantsInfoNew to a .csv file called Participants Information in E:/Work/Multicultural Business Institute/R/R_FBE2021/Meeting 2 folder
    ##the structure: write.csv(nameDataFrameToExport, "locationToExportTo/name.csv")
    ###!!!when writing the location/path use "/". Ubuntu and Mac users: copies the path with "/". Windows users: copies the path with "\" - change it in R.


setwd("E:/Work/Multicultural Business Institute/R/R_FBE2021/Meeting 2") #sets the working directory. all the files exported/imported following this line, are exported to/imported from this location
write.csv(participantsInfoNew, "Participants Information.csv") #exports the participantsInfoNew to a .csv file called Participants Information

#________________________________________
#BAR CHART.PIE CHART. HISTOGRAM.
#*Graphical representations of attributive statistical variables

#*EXERCISE 7: Import the <<Participants Information.csv>> into R.
participants=read.csv("Participants Information.csv") #imports the participants from Participants Information.csv file (from working directory) and stores it in data frame participants
View(participants)


#*EXERCISE 8: Plot the variable <<age>> on a suitable graph. -> BAR CHART (Quantitative Discrete Variable)
##Step 1: Compute the absolute frequencies of <<age>>:
ageFreq=table(participants$age) #the function table sorts the values of vector participants$age and counts their frequency

##Step 2: Plot data:
barplot(ageFreq) #plots a bar chart with vertical bars
barplot(ageFreq, main="Distribution of participants based on age", xlab="age", ylab="no. participants")
#plots a bar chart with:
    ##verical bars
    ##title: Distribution of participants based on age
    ##name oX axis: age
    ##name oY axis: no. part
```

```r

barplot(ageFreq, main="Distribution of participants based on age", xlab="no. participants", ylab="age", horiz=TRUE)
#plots a bar chart with:
    ##horizontal bars
    ##title: Distribution of participants based on age
    ##name oX axis: no. part
    ##name oY axis: age

#*EXERCISE 9: Save your graph in a .png file in your working directory.
png("age.png") #the name of your file will be age with the extension .png
  barplot(ageFreq, main="Distribution of participants based on age", xlab="age", ylab="no. participants")
dev.off()


#*EXERCISE 10: Save your graph in a .pdf file in your working directory.
pdf("age.pdf") #the name of your file will be age with the extension .pdf
  barplot(ageFreq, main="Distribution of participants based on age", xlab="age", ylab="no. participants")
dev.off()

#EXERCISE_POINT_3: Search for other formats you can save your plot in.

#*EXERCISE 11: Plot the variable <<pass>> on a suitable graph.->PIE CHART (Qualitative Nominal Variable)
##Step 1: Compute the absolute frequencies of <<pass>>:
passFreq=table(participants$pass) #the function table sorts the values of vector participants$eye and counts their frequency
View(passFreq)
##Step 2: Plot data:
pie(passFreq)   #plots a pie chart
pie(passFreq, main="Distribution of participants based on pass", col=c("red", "green"), label=c("FAILED", "PASSED"))
#plots a pie chart:
    ##title: Distribution of participants based on pass
    ##FAILED-> COLOUR RED (I know this because failed is the first one in the passFreq table)
    ##PASSED->COLOUR GREEN (I know this because failed is the second one in the passFreq table)

#EXERCISE_POINT_4: Add % to pie chart slices. Add a colour legend instead of labels.

#*EXERCISE 12: Save your graph in a .png file in your working directory.
png("pass.png")
  pie(passFreq, main="Distribution of participants based on pass", col=c("red", "green"), label=c("FAILED", "PASSED"))
dev.off()

#*EXERCISE 13: Generate a random variable <<wage>> of 100 Romanians.  Normal distribution, Mean=3300, sd=1000.
set.seed(4114) #this will allow for reproducibility (I fix my "hat" to be 4114 (my "map" in Minecraft))
wage=rnorm(100, mean=3300, sd=1000) #randomly generates a normal distribution of 100 numbers with mean 3300 and sd 1000. the numbers are drawn from my "hat" 4114

#*EXERCISE 14: Plot <<wage>> on a suitable graph.
hist(wage)  #plots a histogram of wage
hist(wage, main="Distribution of wage", xlab="wage", ylab="no. people")
#plots a histogram with:
##title: income distribution
##name oX axis: wage
##name oY axis: no. people

#*EXERCISE 15: Save your graph in a .png file in your working directory.
png("wage.png")
  hist(wage, main="Distribution of wage", xlab="income", ylab="no. people")
dev.off()
```