**Using R software**

Exercises: correction

# 1. Using R software as a simple calculator

**Exercise n°1:**

a). According to your weight (in kilograms) and height (in metres):

```
> 70 / 1.84^2
[1] 20.67580
```

b).

```
> myWeight <- 70
> myHeight <- 1.84
> myWeight / myHeight^2
[1] 20.67580
```

**Exercise n°2:**

```
> myBudget <- 300
```

a).

```
> myTotal <- 260 + 30 + 60
> myTotal
[1] 350
> myTotal < myBudget
[1] FALSE
```

Completion of the purchase: no, because the total is higher than my budget.

b).

```
> 260 + 30
[1] 290
> (260 + 30) < myBudget
[1] TRUE
```

Completion of the purchase: yes, because the total of these 2 items is lower than my budget.

c).

```
> myDiscount <- (260*30/100) + (30*30/100) + (60*30/100)
> myDiscount
[1] 105
> myExpense <- myTotal - myDiscount
> myExpense
[1] 245
```

The amount of the discount is 105 euros.

The total cost during the sales period will then be 245 euros, which is more attractive than the offer made today.

d).

```
> myBudget - myExpense
[1] 55
> myBudget <- myBudget - myExpense
> myBudget
[1] 55
```

After the purchase, I have 55 euros left

**Exercise n°3:**

a).

```
> ccont <- c(11, 13, 15.5, 12, 8, 9, 13, 16)
> exam <- c(8.5, 14, 15, 10, 12, 13, 14, 17)
> ccont
[1] 11.0 13.0 15.5 12.0  8.0  9.0 13.0 16.0
> exam
[1]  8.5 14.0 15.0 10.0 12.0 13.0 14.0 17.0
```

b). We will continue to work on the data vectors. The scores for the continuous assessment should be weighted by 0.4 (counts for 40% of the final score) and the scores for the examination by 0.6 (counts for 60% of the final score).

```
> tu <- 0.4*ccont + 0.6*exam
> tu
[1]  9.5 13.6 15.2 10.8 10.4 11.4 13.6 16.6
```

c). An element of a vector is accessed by indicating the number of the desired element in square brackets:

```
> ccont[6]
[1] 9
> exam[6]
[1] 13
> tu[6]
[1] 11.4
```

d). Continuous assessment mean (either calculated as the sum of the scores divided by the number of scores, or directly by the **`mean`** function):

```
> sum(ccont) / 8
[1] 12.1875
> mean(ccont)
[1] 12.1875
```

Exam mean:

```
> sum(exam) / 8
[1] 12.9375
> mean(exam)
[1] 12.9375
```

TE mean:

```
> sum(tu) / 8
[1] 12.6375
> mean(tu)
[1] 12.6375
```

Highest score:

```
> max(tu)
[1] 16.6
```

Lowest score:

```
> min(tu)
[1] 9.5
```

## 2. Using R software for descriptive analysis

**Exercise n°4:**

a). Quantitative variables (features): id, age, nbdrugs.
Qualitative variables (features): agecl, sex, sitFamily, wayolife, homeassist, fall, CV, Psy, Antidiabetic, otherdrugs, autowalk.

Note: wayolife, CV, Psy, Antidiabetic, otherdrugs are numerical variables that take the values 0 or 1 which correspond to the qualitative values 'No' and 'Yes'.

After transforming the ".xlsx" file to ".csv" format, it must be imported to an R directory using the following command:

```
> fallers <- read.csv2(" … fallers.csv", header=TRUE)

> str(fallers)
'data.frame':   153 obs. of  14 variables:
 $ id           : int  3 6 7 10 12 19 21 26 29 40 ...
 $ age          : int  87 81 78 75 77 96 78 66 81 67 ...
 $ agecl        : chr  "75+" "75+" "75+" "75+" ...
 $ sex          : chr  "Woman" "Woman" "Woman" "Woman" ...
 $ sitFamily    : chr  "Widower" "Widower" "Widower"
"Widower" ...
 $ wayolife     : chr  "Alone" "Alone" "Alone" "Alone" ...
 $ homeassist   : int  0 0 0 0 0 1 0 0 0 0 ...
 $ fall         : chr  "Yes" "No" "No" "No" ...
 $ nbdrugs      : int  6 6 8 2 13 3 10 1 8 3 ...
 $ CV           : int  1 1 1 0 1 0 1 0 1 0 ...
 $ Psy          : int  0 1 0 1 1 0 1 1 0 0 ...
 $ Antidiabetic: int  1 0 1 1 0 1 0 0 1 0 ...
 $ otherdrugs   : int  1 0 1 0 0 1 0 0 1 1 ...
 $ autowalk     : chr  "Assist" "Yes" "Yes" "Yes" ..
```

b). Recoding:

```
# Recoding of variables in character format
# into categorical features
fallers$agecl <- as.factor(fallers$agecl)
fallers$sex <- as.factor(fallers$sex)
fallers$sitFamily <- as.factor(fallers$sitFamily)
fallers$wayolife <- as.factor(fallers$wayolife)
fallers$fall <- as.factor(fallers$fall)
fallers$autowalk <- as.factor(fallers$autowalk)

# Reconding of variables in integer format
# into categorical features with a label
# except for nbdrgugs wich is continuous
fallers$homeassist <- factor(fallers$homeassist,
      labels=c("No","Yes"), levels=c(0, 1))
fallers$autowalk <- factor(fallers$autowalk,
      levels=c("Yes", "Assist", "No"),
      labels=c("Yes", "Assist", "No"))
fallers$CV <- factor(fallers$CV,
      levels=c(0, 1),
```

```
      labels=c("No","Yes"))
fallers$Psy <- factor(fallers$Psy,
      levels=c(0, 1),
      labels=c("No","Yes"))
fallers$Antidiabetic <- factor(fallers$Antidiabetic,
      levels=c(0, 1),
      labels=c("No","Yes"))
fallers$otherdrugs <- factor(fallers$otherdrugs,
      levels=c(0, 1),
      labels=c("No","Yes"))
```

c). The data.frame must be imported attached to working environment using the following command:

```
> attach(fallers)
```

Absolute frequencies: 2 commands are possible

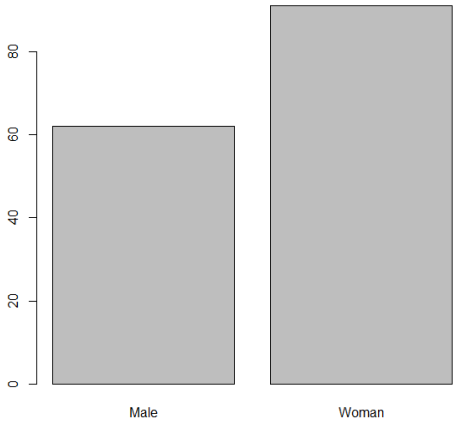
```
> summary(sex)
Male Woman
   62    91
> table(sex)

sex

 Male Woman

   62    91
```

Relative frequencies: absolute frequencies divided by the total population, which is obtained by determining the number of elements (command **`length`**) of the sex vector

```
> summary(sex)/length(sex)
     Male     Woman
0.4052288 0.5947712
> table(sex)/length(sex)
sex
     Male     Woman
0.4052288 0.5947712
> prop.table(table(sex))
sex
      Man     Woman
0.4052288 0.5947712
```

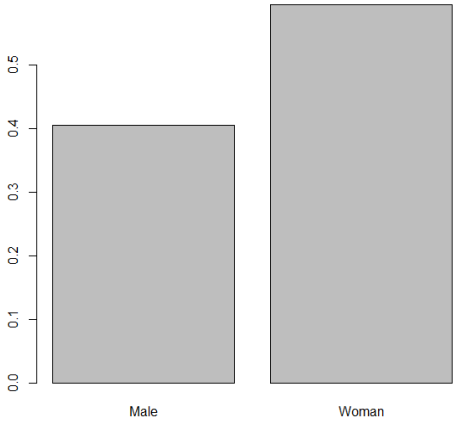
Histogram of absolute frequencies:

```
> barplot(summary(sex))
```



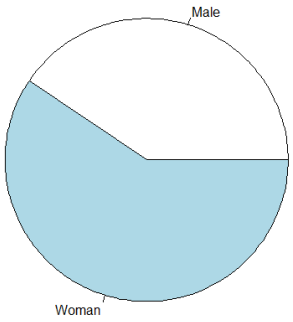

Histogram of relative frequencies:

```
> barplot(summary(sex)/length(sex))
```




Pie chart représentation:

```
> pie(summary(sex))
```



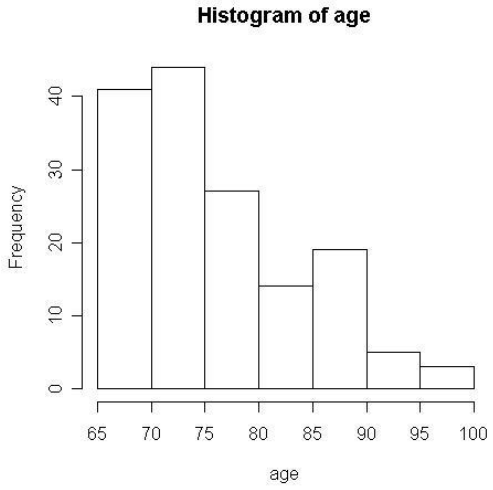

d).

```
> summary(age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  65.00   70.00   75.00   76.29   81.00  100.00
```

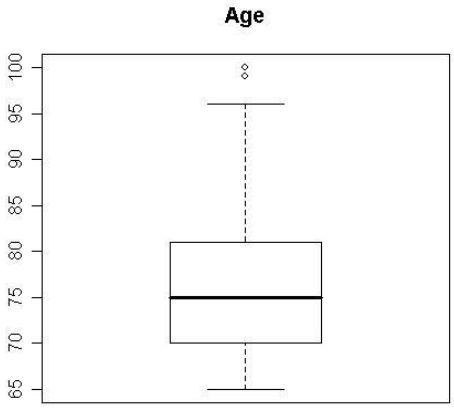
Histogram of frequencies:

```
> hist(age)
```



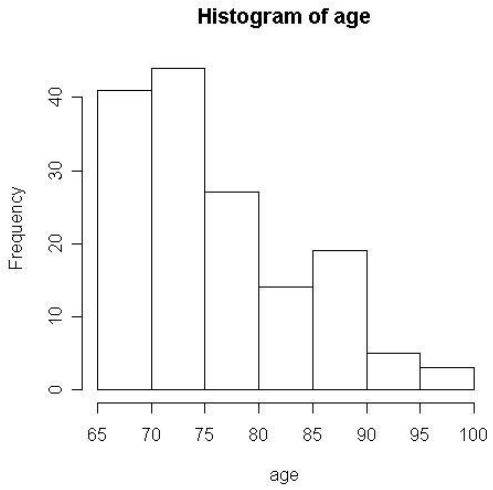

Representation in the form of a box-plot:

```
> boxplot(age, main = "Age")
```




e). Use **`?hist`** to understand how this command works. The **`breacks`** argument should be used to build your own intervals.

```
> hist(age, breaks=c(65, 70, 75, 80, 85, 90, 95, 100))
```




**Exercise n°5:**

a). This is a quantitative variable that we want to describe in terms of the levels of a qualitative variable.

The **`summary`** command included in **`by`** provides statistical summaries of age as a function of the level of the fall variable.

```
> by(age, fall, summary)
fall: No
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  65.00   69.00   74.00   75.58   80.50  100.00
------------------------------------------------
fall: Yes
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  66.00   75.00   77.00   78.76   84.75   96.00
```

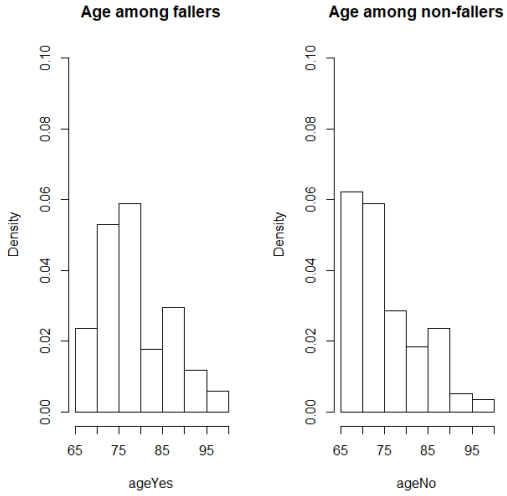
Another possibility: construct the age vectors as a function of the level of the **`fall`** variable, then use the **`summary`** command:

```
> ageNo <- age[fall=="No"]
> summary(ageNo)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  65.00   69.00   74.00   75.58   80.50  100.00
> ageYes <- age[fall=="Yes"]
> summary(ageYes)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  66.00   75.00   77.00   78.76   84.75   96.00
```

b).

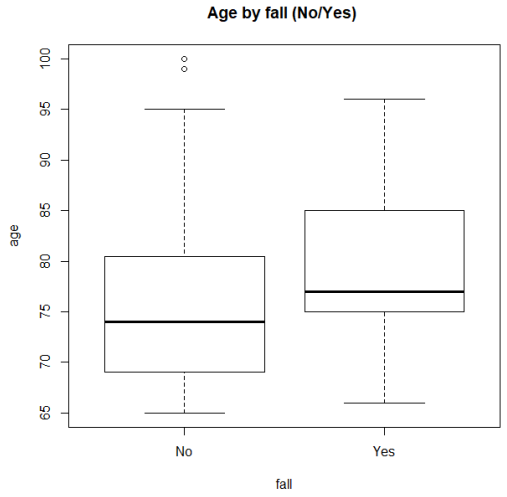
```
> ageYes <- age[fall=="Yes"]
> ageNo <- age[fall=="No"]

> old.par <- par(no.readonly = TRUE)
> par(mfrow = c(1, 2))
> hist(ageYes, breaks = seq(65, 100, 5), freq = FALSE,
+      ylim = c(0, 0.1), main = "Age among fallers")
> hist(ageNo, breaks = seq(65, 100, 5), freq = FALSE,
+      ylim = c(0, 0.1), main = " Age among non-fallers ")
> par(mfrow = c(1, 1))
```

```
> par(old.par)
```




c).

```
> boxplot(age ~ fall, main = "Age by fall (No/Yes) ")
```




**Exercise n°6:**

a). These are two qualitative variables whose association is described by a contingency table:

```
> table(wayolife, fall)
          fall
wayolife   No Yes
  Alone    26   9
  Notalone 93  25
```

b). The first step is to create the vectors that fall among people not living alone and those living alone:

```
> fallNotalone <- fall[wayolife=="Notalone"]
> fallAlone <- fall[wayolife=="Alone"]
```

In a second step, make tables of the relative frequencies of falls among people not living alone and then among those living alone:

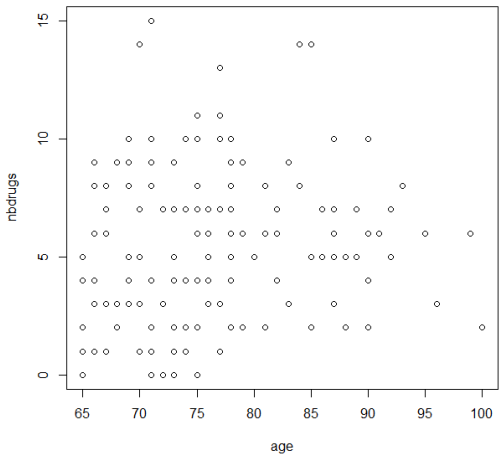
```
> table(fallNotalone)/length(fallNotalone)
FallNotalone
       No       Yes
0.7881356 0.2118644
```

```
> table(fallAlone)/length(fallAlone)
fallAlone
       No       Yes
0.7428571 0.2571429
```

**Exercise n°7:**

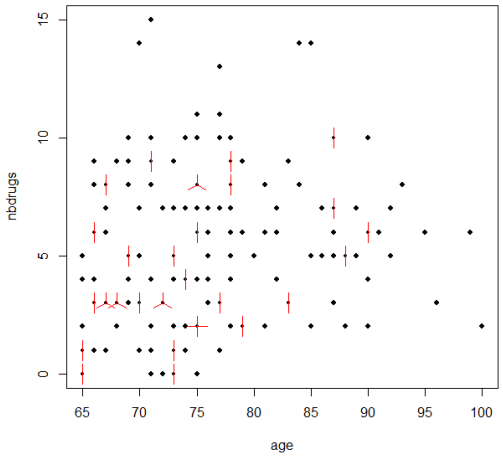
a). These are 2 quantitative variables. The graphical representation is a scatter plot:

```
> plot(age, nbdrugs)
```




Another function can be used: **`sunflowerplot`** which gives a graph similar to that obtained with the **`plot`** command but the superimposed points are drawn in the shape of flowers whose number of petals represents the number of points.

```
> sunflowerplot(age, nbdrugs)
```




b). These are 2 quantitative variables. One possible statistic is the correlation coefficient:

```
> cor(age, nbdrugs, method=c("pearson"))
[1] 0.1603727
> cor(age, nbdrugs, method=c("spearman"))
[1] 0.2057321
```