LMAFY1101 - Solutions - Série 1

Utilisation de R

Exercice 1

1.

```
x <- c(-2, -1, 0, 1, 2, 3)  # ou
## x <- seq(-2, 3, by = 1)
x
```

```
[1] -2 -1  0  1  2  3
```

2.

```
x + 2
```

```
[1] 0 1 2 3 4 5
```

3.

```
y <- rep(5, 5)
y
```

```
[1] 5 5 5 5 5
```

```
z <- rep(c(-1, 2, 1), c(5, 1, 8))
z
```

```
 [1] -1 -1 -1 -1 -1  2  1  1  1  1  1  1  1  1
```

4.

```
y + c(0, z)
```

```
 [1] 5 4 4 4 4 4 7 6 6 6 6 6 6 6 6
```

```
y + z
```

```
Warning in y + z: longer object length is not a multiple of shorter object
length
```

```
 [1] 4 4 4 4 4 7 6 6 6 6 6 6 6 6
```

5.

```
z <- z[-length(z)]
z
```

```
 [1] -1 -1 -1 -1 -1  2  1  1  1  1  1  1  1
```

6.

```
c(x, y)
```

```
 [1] -2 -1  0  1  2  3  5  5  5  5  5
```

7.

```
x[c(1, 2, 3)] <- c(7, 6, 5)
x
```

```
[1] 7 6 5 1 2 3
```

8.

```
sort(x)
```

```
[1] 1 2 3 5 6 7
```

```r
order(x)
```

```
[1] 4 5 6 3 2 1
```

```r
x[order(x)]
```

```
[1] 1 2 3 5 6 7
```

9.

```r
w <- z[z > 0]   # ou
## w <- z[ifelse(z > 0, TRUE, FALSE)]
w
```

```
[1] 2 1 1 1 1 1 1 1
```

Exercice 2

1.

```r
taille <- c(160, 176, 161, 165, NA, 168, 161, 174, 161, 159,
  164, 169, 163, 163, NA, 172, 165, 164, 170, 163, 169, 184)
```

2.

```r
length(taille)
```

```
[1] 22
```

3.

```r
any(is.na(taille))
```

```
[1] TRUE
```

4.

```r
sum(is.na(taille))
```

```
[1] 2
```

5.

```r
mean(taille, na.rm = TRUE)
```

```
[1] 167
```

6.

```r
summary(taille)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
    159     162     164     167     169     184       2 
```

7.

```r
taille_ok <- taille[!is.na(taille)]  # ou
## taille_ok <- na.omit(taille)
taille_ok
```

```
 [1] 160 176 161 165 168 161 174 161 159 164 169 163 163 172 165 164 170 163 169
[20] 184
```

8.

```r
taille_ok[-(1:10)]
```

```
 [1] 169 163 163 172 165 164 170 163 169 184
```

9.

```r
taille_ok <- taille_ok/100
taille_ok
```

```
 [1] 1.60 1.76 1.61 1.65 1.68 1.61 1.74 1.61 1.59 1.64 1.69 1.63 1.63 1.72 1.65
[16] 1.64 1.70 1.63 1.69 1.84
```

10.

```r
length(taille_ok[taille_ok > 1.65])
```

```
[1] 8
```

11.

```r
length(taille_ok[taille_ok >= 1.6 & taille_ok <= 1.7])
```

```
[1] 15
```

Exercice 3

```r
diam.cylindre <- c(4.03, 4.05, 3.96, 4.09, 4.28, 4.04, 4.18,
  4.23, 4.14, 4.12, 4.03, 3.94, 4.02, 4.08, 4.13, 4.04, 3.93,
  4.08, 4.37, 4.07, 4.11, 4.03, 4, 3.97, 4.01, 4.09, 4.06,
  3.92, 4.19, 3.96, 4.48, 4.24, 4.06, 3.98)
```

1.

```r
c(mean(diam.cylindre), sd(diam.cylindre), sd(diam.cylindre)/mean(diam.cylindre))
```

```
[1] 4.0856 0.1241 0.0304
```

2.

```r
summary(diam.cylindre)[-4]
```

```
   Min. 1st Qu.  Median 3rd Qu.    Max. 
   3.92    4.01    4.06    4.13    4.48 
```

3.

```r
quantile(diam.cylindre, probs = c(0.3, 0.7))
```

```
 30%  70% 
4.03 4.11 
```

4.

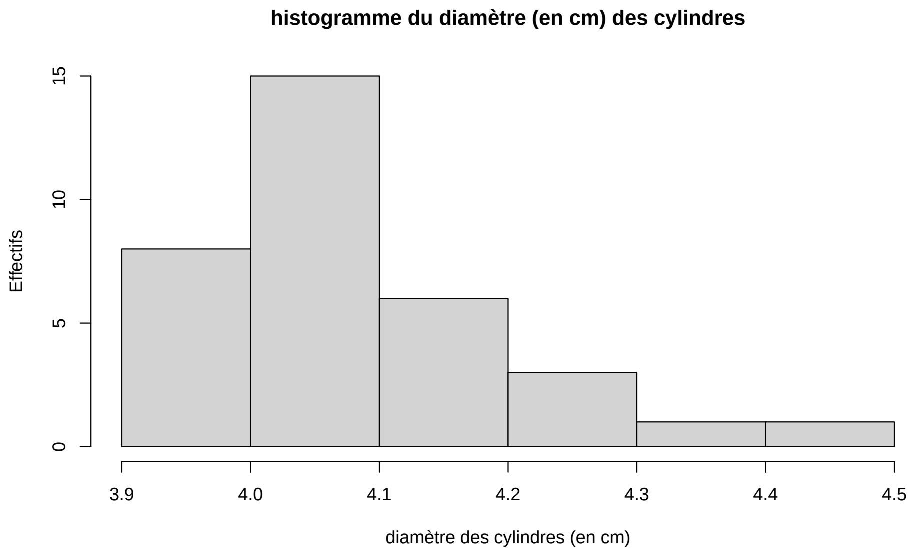
```r
diam.cylindre2 <- c(4.03, 4.05, 3.96, 4.09, 4.28, 4.04, 4.68,
  4.73, 4.64, 4.62, 4.03, 3.94, 4.02, 4.08, 4.13, 4.04, 3.93,
  4.08, 4.37, 4.07, 4.11, 4.03, 4, 3.97, 4.01, 4.09, 4.06,
  3.92, 4.19, 3.96, 4.48, 4.24, 4.06, 3.98)

c(mean(diam.cylindre2), median(diam.cylindre2))
```

```
[1] 4.14 4.06
```

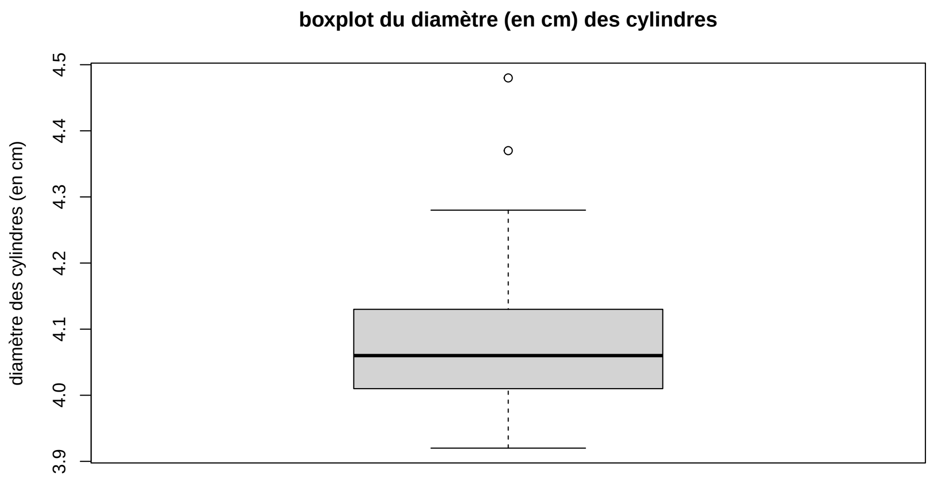
5.

```r
hist(diam.cylindre, main = "histogramme du diamètre (en cm) des cylindres",
  xlab = "diamètre des cylindres (en cm)", ylab = "Effectifs")
```



6.

```r
boxplot(diam.cylindre, main = "boxplot du diamètre (en cm) des cylindres",
  ylab = "diamètre des cylindres (en cm)")
```



Exercice 5

1.

```
Noms <- c("Victor", "Sandrine", "Jonathan", "Marie")
Ages <- c(4, 7, 6, 4)
Tailles <- c(110, 122, 125, 118)
```

2.

```
data <- data.frame(Noms = Noms, Ages = Ages, Tailles = Tailles)
```

3.

```
data[order(Noms), ]
```

```
      Noms Ages Tailles
3 Jonathan    6     125
4    Marie    4     118
2 Sandrine    7     122
1   Victor    4     110
```

4.

```
subset(data, subset = Ages == min(Ages))
```

```
    Noms Ages Tailles
1 Victor    4     110
4  Marie    4     118
```

5.

```
Noms[order(Tailles, decreasing = TRUE)]
```

```
[1] "Jonathan" "Sandrine" "Marie"    "Victor"
```

Exercice 6

1.

```
daf <- data.frame(col1 = c(10.9, 12.4, 11.9, 13.2, 11.1), col2 = c(8,
  4, 2, 6, 10), col3 = c("Anne", "Michel", "Dominique", "Camille",
  "Stéphane"), col4 = c(1, 1, 2, 2, 1))
```

2.

```
View(daf)
```

3.

```
dim(daf)
```

```
[1] 5 4
```

```
nrow(daf)
```

```
[1] 5
```

```
ncol(daf)
```

```
[1] 4
```

4.

```r
str(daf)
```

```
'data.frame':	5 obs. of  4 variables:
 $ col1: num  10.9 12.4 11.9 13.2 11.1
 $ col2: num  8 4 2 6 10
 $ col3: chr  "Anne" "Michel" "Dominique" "Camille" ...
 $ col4: num  1 1 2 2 1
```

5.

```r
daf[3, 2]
```

```
[1] 2
```

6.

```r
summary(daf)
```

```
      col1            col2          col3                col4    
 Min.   :10.9   Min.   : 2   Length:5           Min.   :1.0  
 1st Qu.:11.1   1st Qu.: 4   Class :character   1st Qu.:1.0  
 Median :11.9   Median : 6   Mode  :character   Median :1.0  
 Mean   :11.9   Mean   : 6                      Mean   :1.4  
 3rd Qu.:12.4   3rd Qu.: 8                      3rd Qu.:2.0  
 Max.   :13.2   Max.   :10                      Max.   :2.0  
```

7.

```r
daf[, 1]  # ou
## daf$col1
## daf["col1"]
## subset(daf, select = col1)
```

```
[1] 10.9 12.4 11.9 13.2 11.1
```

8.

```r
names(daf)  # ou
## colnames(daf)
```

```
[1] "col1" "col2" "col3" "col4"
```

```
rownames(daf)
```

```
[1] "1" "2" "3" "4" "5"
```

9.

```
names(daf)[3] <- "Nom"
```

10.

```
str(daf[, 4])  # ou
## class(daf[, 4])
```

```
 num [1:5] 1 1 2 2 1
```

```
daf[, 4] <- factor(daf[, 4])  # ou
## daf <- transform(daf, col4 = factor(col4))
```

11.

```
levels(daf$col4) <- c("non", "oui")
daf$col4
```

```
[1] non non oui oui non
Levels: non oui
```

12.

```
daf$col2 <- daf$col2/10  # ou
## daf <- transform(daf, col2 = col2/10)
daf
```

```
  col1 col2       Nom col4
1 10.9  0.8      Anne  non
2 12.4  0.4    Michel  non
3 11.9  0.2 Dominique  oui
4 13.2  0.6   Camille  oui
5 11.1  1.0  Stéphane  non
```

13.

```r
daf[daf$col2 > 0.5, ]  # ou
## subset(daf, subset = col2 > 0.5)
```

```
  col1 col2      Nom col4
1 10.9  0.8     Anne  non
4 13.2  0.6  Camille  oui
5 11.1  1.0 Stéphane  non
```

14.

```r
daf[daf$col1 > 11.5, "Nom"]  # ou
## subset(daf, subset = col1 > 11.5, select = Nom)
```

```
[1] "Michel"    "Dominique" "Camille"
```

15.

```r
nrow(subset(daf, subset = col1 >= 11 & col1 <= 12))  # ou
## daf |>
##   subset(subset = col1 >= 11 & col1 <= 12) |>
##   nrow()
```

```
[1] 2
```

16.

```r
sum(subset(daf, subset = col4 == "oui", select = col2))  # ou
## daf |>
##   subset(subset = col4 == "oui", select = col2) |>
##   sum()
```

```
[1] 0.8
```

Exercice 7

2.

Via le menu de RStudio :
`Session > Set Working Directory > Choose Directory...`

4.

Vous pouvez utiliser le volet `Import Dataset` de RStudio ou tapez le code suivant qui suppose que “data” est votre répertoire de travaille R.

```
climate <- read.csv("data/Ex5_climate.csv", sep = ";")
ls()
```

5.

```
save(climate, file = "data/climate.rda")
rm(climate)
ls()
```

6.

```
load("data/climate.rda")
ls()
```

7.

```
rm(list = ls())
```

Exercice 8

1.

```
library(ggplot2)
```

2.

```
diamonds <- data.frame(diamonds)
head(diamonds)
```

```
  carat       cut color clarity depth table price    x    y    z
1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

3.

```r
# Nombre d'observations
nrow(diamonds)
```

```
[1] 53940
```

```r
# Nombre de variables
ncol(diamonds)
```

```
[1] 10
```

```r
# Nombre d'observations et Nombre de variables
dim(diamonds)
```

```
[1] 53940    10
```

4.

```r
help(diamonds)
```

5.

```r
str(diamonds)
```

```
'data.frame':   53940 obs. of  10 variables:
 $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
 $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

6.

```r
diamonds <- transform(diamonds, price.euros = price/1.23)
head(diamonds)
```

```
  carat       cut color clarity depth table price    x    y    z price.euros
1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43         265
2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31         265
3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31         266
4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63         272
5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75         272
6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48         273
```

7.

```r
diamonds2 <- subset(diamonds, cut == "Fair")
head(diamonds2)
```

```
    carat  cut color clarity depth table price    x    y    z price.euros
9    0.22 Fair     E     VS2  65.1    61   337 3.87 3.78 2.49         274
92   0.86 Fair     E     SI2  55.1    69  2757 6.45 6.33 3.52        2241
98   0.96 Fair     F     SI2  66.3    62  2759 6.27 5.95 4.07        2243
124  0.70 Fair     F     VS2  64.5    57  2762 5.57 5.53 3.58        2246
125  0.70 Fair     F     VS2  65.3    55  2762 5.63 5.58 3.66        2246
129  0.91 Fair     H     SI2  64.4    57  2763 6.11 6.09 3.93        2246
```

8.

```r
diamonds3 <- subset(diamonds, (cut != "Fair" & price > 10000))
head(diamonds3)
```

```
      carat       cut color clarity depth table price    x    y    z
21929  1.70     Ideal     J     VS2  60.5    58 10002 7.73 7.74 4.68
21930  1.03     Ideal     E    VVS2  60.6    59 10003 6.50 6.53 3.95
21931  1.23 Very Good     G    VVS2  60.6    55 10004 6.93 7.02 4.23
21932  1.25     Ideal     F     VS2  61.6    55 10006 6.93 6.96 4.28
21933  2.01 Very Good     I     SI2  61.4    63 10009 8.19 7.96 4.96
21934  1.21 Very Good     F     VS1  62.3    58 10009 6.76 6.85 4.24
      price.euros
21929        8132
21930        8133
21931        8133
21932        8135
21933        8137
21934        8137
```