TP R 1: Solutions des exercices

3.1 Vecteurs, matrices, listes

1.a Exercice 2.2 livre V. Goulet

```
x <- c(1,18,2,1,5,2,6,1,12,3,13,8,20,1,5,7,7,4,14,10)
# a
x[2]
# b
x[1:5]
# c
x[x>14]
x[which(x>14)]
# d
x[-c(6,10,12)]
# Question supplémentaire
x[1] <- NA
x[x>5] # attention aux données manquantes!
x[which(x>5)]
```

1.b Exercice 2.3 livre V. Goulet

```
x <- matrix(sample(1:100,70), 7, 10)
# a
x[4,3]
# b
x[6,]
# c
x[,c(1,4)]
# d
x[x[,1]>50,]
x[which(x[,1]>50),]
```

1.c Exercice 2.1 livre V. Goulet

```
# a
(x <- list(1:5, data=matrix(1:6, nrow=2), rep(0,3), test=rep(FALSE,4)))
# b
names(x)
# c
mode(x$test)
length(x$test)
# d
dim(x[[2]])
# e
x[[2]][2:3]
# f
```

```r
x[[3]] <- 3:8
x
```

1.d Exercice 3.1 livre V. Goulet

```r
# a
rep(c(0,6),3)
# b
seq(from=1,to=10,by=3)
# c
rep(1:3,4)
# d
rep(1:3, 1:3)
# d
rep(1:3,1:3)
# e
rep(1:3,3:1)
# f
seq(1,10,length=3)
# g
rep(1:3,rep(4,3))
```

1.e Exercice 3.2 livre V. Goulet

```r
# a
11:20/10
# seq(1.1,2,by=.1)
# b
0:9*2+1
# seq(1,19,by=2)
# c
rep(-2:2,2)
# d
rep(-2:2,rep(2,5))
# e
1:10*10
```

1.f Exercice 3.8 livre V. Goulet

```r
M <- matrix(rpois(10*7,lambda=8),nrow=10)
# a
apply(M,1,sum)
# b
apply(M,2,mean)
#c
m <- M[1:3,1:3]
max(m)
#d
M[apply(M,1,mean)>7,]
```

2

```
B <- matrix(c(3,-4,0,0,-1,0,0,-8,3),ncol=3)
# déterminant
det(B)
# valeur et vecteurs propres
e <- eigen(B)
# vérification:
(e.vectors <- e$vectors)
(e.values <- e$values)
B %*% e.vectors
e.values[1] * e.vectors[,1]
e.values[2] * e.vectors[,2]
e.values[3] * e.vectors[,3]
# ou, plus rapidement
e.vectors %*% diag(e.values)
# vérification diagonalisation
e.vectors %*% diag(e.values) %*% solve(e.vectors) == B
solve(e.vectors) %*% B %*% e.vectors == diag(e.values)
# inverse
inv.B <- solve(B)
inv.B %*% B
```

3.2 Fonctions et structures de contrôle

1.a Exercice 4.2 livre V. Goulet

```
mp <- function(x,w){
  return(sum(w*x)/sum(w))
}
x <- c(7, 13, 3, 8, 12, 12, 20, 11)
w <- c (0.15, 0.04, 0.05, 0.06, 0.17, 0.16, 0.11, 0.09)
mp(x,w)
```

1.b Exercice 5.1 livre V. Goulet

```
variance <- function(x,biased=FALSE){
  n <- length(x)
  m <- mean(x)
  if(biased == FALSE){
    s2 <- sum((x-m)^2)/(n-1)
    }else{
    s2 <- sum((x-m)^2)/n
    }
  return(s2)
}
x <- rnorm(1000) #1000 valeurs tirés selon la loi normale centrée et réduite
variance(x)
var(x)
variance(x, biased=TRUE)
```

2

On a $f(x) = x^2 - a$. Donc $f'(x) = 2x$ et

$$x_{n+1} = \frac{x_n^2 + a}{2x_n}$$

```
racine <- function(a,x1,n){
  x <- numeric(length=n) # on crée un vecteur numérique de longuer n
  x[1] <- x1
  for(i in 1:n){
  x[i+1] <- (x[i]+a/x[i])/2
  }
  return(x)
}
racine(a=2,x1=3,n=10)
# Attention au point de départ:
racine(a=2,x1=0,n=10)
racine(a=2,x1=-2,n=10)
```

3.3 Importation des données et data frame

```
setwd("~/Desktop/Programmation/data/") #changer avec le répertoire approprié
#1
hep <- read.table('hepatitis.txt',header=TRUE,sep='\t',na.strings='?')
#2
nrow(hep)
names(hep)
head(hep)
is.na(hep$STEROID[4])
#3
attach(hep)
mean(ALBUMIN[SEX == 'male'])
mean(ALBUMIN[SEX == 'female'])
#4.
hep$NYSMP <- (hep$FATIGUE == "yes") + (hep$MALAISE == "yes")
head(hep[,c("FATIGUE","MALAISE","NYSMP")])

# autre stratégie, en exploitant la structure de facteur:
FATIGUE[1:4]
as.numeric(FATIGUE[1:4])
as.numeric(FATIGUE[1:4])-1 # no = 0, yes = 1
hep$NYSMP2 <- as.numeric(FATIGUE)-1 + as.numeric(MALAISE)-1
head(hep[,c("FATIGUE","MALAISE","NYSMP","NYSMP2")])
```