

# correction td2 ISV51

## exercice 1 : recyclage d'éléments

```
u <- c(10,20,30)
v <- 1:9
u+v
```

```
## [1] 11 22 33 14 25 36 17 28 39
```

## exercice 2 : Génération de vecteurs

1. Entiers de 1 à 12 de 2 manières différentes

```
1:12
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12
```

```
seq(1,12)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12
```

```
seq_len(12)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12
```

2. Trois manières de générer  $c(0.5, 1, \dots, 5.0)$

```
seq(0.5,5,by=0.5)
```

```
## [1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

```
1:10/2
```

```
## [1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

```
seq(0.5,5,len=10)
```

```
## [1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

```
sort(c(0.5:4.5,1:5))
```

```
## [1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

3. Multiples de 2 entre 1 et 50

```
seq(2,50,by=2)
```

```
## [1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46  
## [24] 48 50
```

```
x <- 1:50; x[x%%2==0]
```

```
## [1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46  
## [24] 48 50
```

4. Tous les nombres de 1 à 100 qui ne sont pas des multiples de 5

```
x <- 1:100; x[x %% 5 != 0]
```

```
## [1] 1 2 3 4 6 7 8 9 11 12 13 14 16 17 18 19 21 22 23 24 26 27 28  
## [24] 29 31 32 33 34 36 37 38 39 41 42 43 44 46 47 48 49 51 52 53 54 56 57  
## [47] 58 59 61 62 63 64 66 67 68 69 71 72 73 74 76 77 78 79 81 82 83 84 86  
## [70] 87 88 89 91 92 93 94 96 97 98 99
```

5. Contenant 3 fois chacun des 10 chiffres

```
rep(1:10,3)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3  
## [24] 4 5 6 7 8 9 10
```

```
rep(1:10,each=3)
```

```
## [1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6 7 7 7 8 8  
## [24] 8 9 9 9 10 10 10
```

6. Chaîne avec LETTERS:

```
length(rep(LETTERS,1:26))
```

```
## [1] 351
```

7. Créer le vecteur c("individu 1", "individu 2", ..., "individu 100").

```
head(paste("individu", 1:100, sep="."))
```

```
## [1] "individu.1" "individu.2" "individu.3" "individu.4" "individu.5"  
## [6] "individu.6"
```

### exercice 3 : manipulation de séquences

1. Entiers divisibles par 3 parmi les 100 premiers

```
x <- 1:100; x <- x[x %% 3 == 0]
length(x)
```

```
## [1] 33
```

```
sum(x)
```

```
## [1] 1683
```

```
prod(x)
```

```
## [1] 4.827109e+52
```

## 2. Séquence d'ADN

```
alphabet <- c("a","c","g","t")
n <- 1e6
ADN <- sample(alphabet, n, rep=TRUE)
nb.a <- sum(ADN == "a")
nb.c <- sum(ADN == "c")
nb.g <- sum(ADN == "g")
nb.t <- sum(ADN == "t")
table(ADN)
```

```
## ADN
##      a      c      g      t
## 249561 250609 249672 250158
```

```
ind.t <- which(ADN == "t")
```

## 3. Séquence encore

```
set.seed(1)
S <- sample(1:100)
which.min(S)
```

```
## [1] 27
```

```
which.max(S)
```

```
## [1] 87
```

```
x <- sort(S)
y <- sort(S, decreasing=TRUE)
c(x,y)[-which(diff(c(x,y)) == 0)]
```

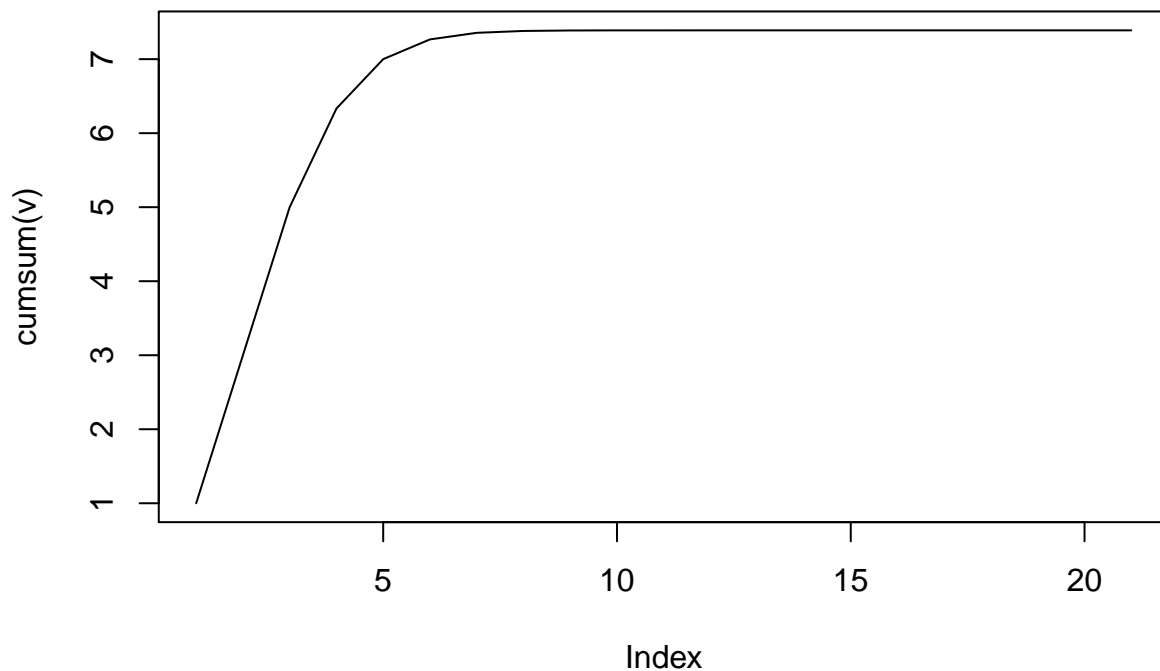
```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
## [18] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
## [35] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
## [52] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
## [69] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
## [86] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 99 98
## [103] 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81
## [120] 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64
## [137] 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47
## [154] 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30
## [171] 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13
## [188] 12 11 10 9 8 7 6 5 4 3 2 1
```

#### 4. Exponentielle

```
v <- 2^(0:20)/factorial(0:20)
sum(v[which(v > 1e-8)])
```

```
## [1] 7.389056
```

```
plot(cumsum(v), type="l")
```



exercice 4 : jeux de hasard

## ex-

##### 1. pile ou face

```
tirage <- sample(c("pile", "face"), 1000, rep=TRUE)
sum(tirage == "pile") - sum(tirage == "face")
```

```
## [1] 34
```

## 2. Lancers de dé

```
## valeurs associées au nombre valeur du dé
issues <- c(-2.5,-2.5,-2.5,1,3,3)
mean(sample(issues,1e5,rep=TRUE)) ## moyenne empirique
```

```
## [1] -0.08253
```

```
sum(issues*1/6) # espérance
```

```
## [1] -0.08333333
```

## 5. Couleurs

```
couleurs <- c("rouge","vert","bleu","jaune","rose","violet")
choose(6,3); combn(couleurs,4)
```

```
## [1] 20
```

```
##      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]
## [1,] "rouge" "rouge" "rouge" "rouge" "rouge" "rouge" "rouge" "rouge"
## [2,] "vert"  "vert"  "vert"  "vert"  "vert"  "vert"  "bleu"  "bleu"
## [3,] "bleu"  "bleu"  "bleu"  "jaune" "jaune" "rose"  "jaune" "jaune"
## [4,] "jaune" "rose"  "violet" "rose"  "violet" "violet" "rose"  "violet"
##      [,9]  [,10]  [,11]  [,12]  [,13]  [,14]  [,15]
## [1,] "rouge" "rouge" "vert"  "vert"  "vert"  "vert"  "bleu"
## [2,] "bleu"  "jaune" "bleu"  "bleu"  "bleu"  "jaune" "jaune"
## [3,] "rose"  "rose"  "jaune" "jaune" "rose"  "rose"  "rose"
## [4,] "violet" "violet" "rose"  "violet" "violet" "violet" "violet"
```

```
mes.couleurs <- sample(couleurs, 100, rep=TRUE)
primaires <- c("rouge", "jaune", "bleu")
sum(mes.couleurs %in% primaires)
```

```
## [1] 55
```

```
length(which(mes.couleurs %in% primaires))
```

```
## [1] 55
```

```
mes.couleurs[mes.couleurs %in% primaires]
```

```
## [1] "bleu" "jaune" "bleu" "bleu" "jaune" "rouge" "jaune" "rouge"
## [9] "bleu" "jaune" "jaune" "bleu" "bleu" "rouge" "rouge" "bleu"
## [17] "rouge" "jaune" "jaune" "bleu" "jaune" "bleu" "jaune" "bleu"
## [25] "rouge" "bleu" "jaune" "jaune" "rouge" "rouge" "rouge" "rouge"
## [33] "rouge" "rouge" "rouge" "rouge" "bleu" "bleu" "rouge" "rouge"
## [41] "jaune" "jaune" "jaune" "bleu" "jaune" "rouge" "jaune" "jaune"
## [49] "rouge" "jaune" "bleu" "rouge" "jaune" "jaune" "rouge"
```

## exercice 5 : résumé numérique

```
grp1 <- c(14.40 , 13.70 , 14.20 , 17.30 , 13.90 , 13.60 , 15.40 , 10.80 , 12.20 , 13.60)  
grp2 <- c(14.00 , 15.90 , 16.90 , 14.10 , 13.80 , 20.30 , 16.00 , 15.30 , 16.10 , 15.90)
```

```
mean(grp1); mean(grp2)
```

```
## [1] 13.91
```

```
## [1] 15.83
```

```
median(grp1); median(grp2)
```

```
## [1] 13.8
```

```
## [1] 15.9
```

```
var(grp1); var(grp2)
```

```
## [1] 2.985444
```

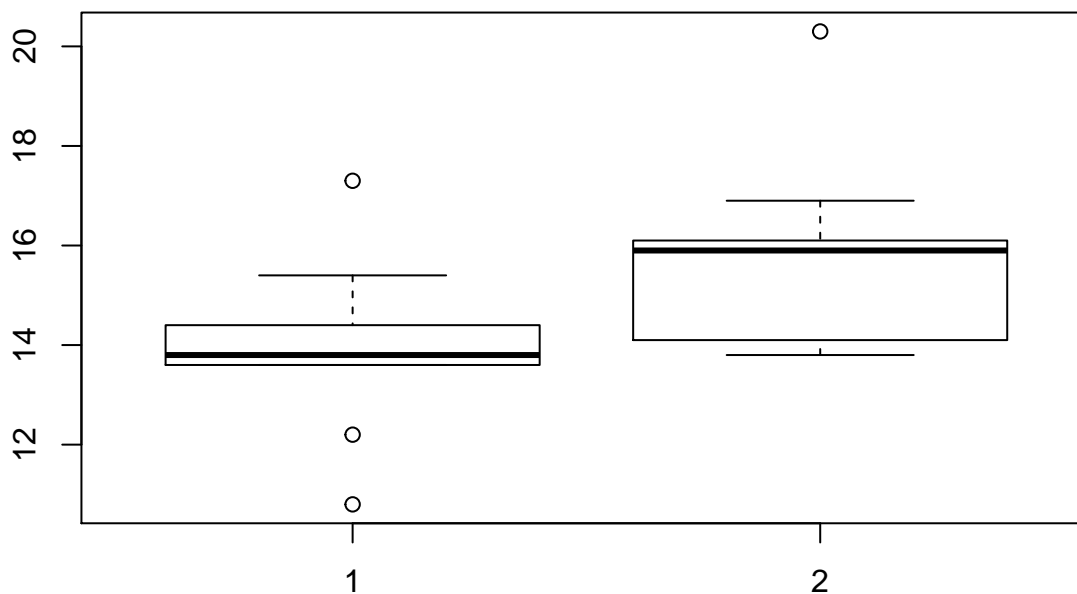
```
## [1] 3.553444
```

```
sd(grp1); sd(grp2)
```

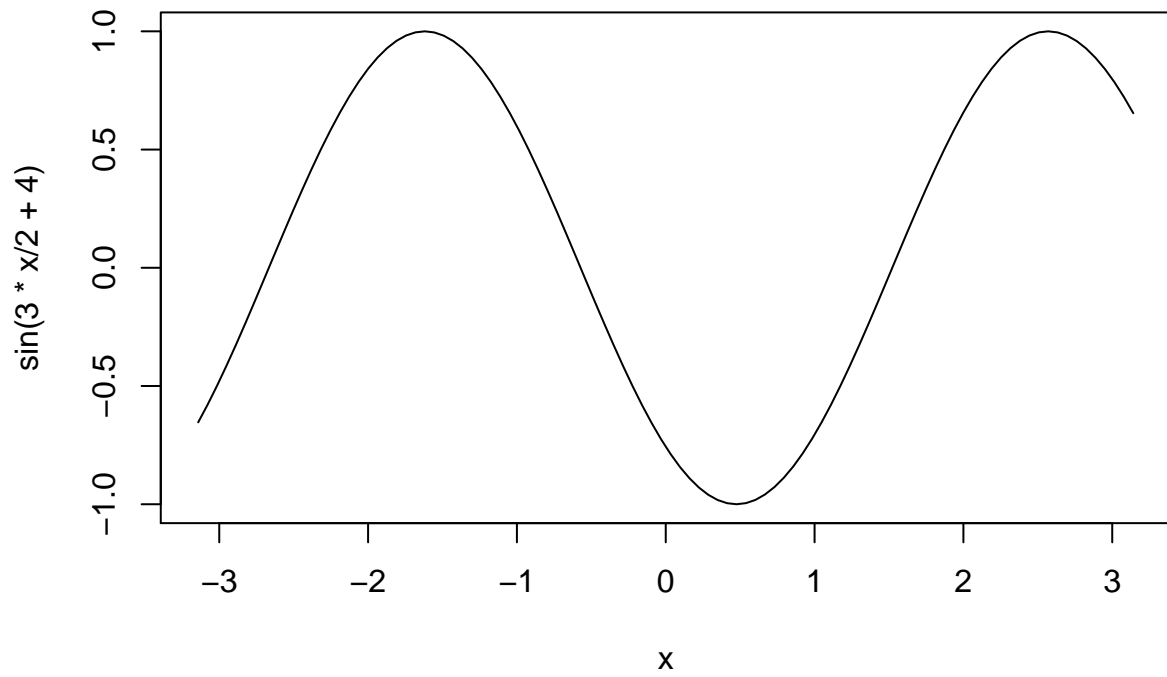
```
## [1] 1.727844
```

```
## [1] 1.885058
```

```
boxplot(grp1,grp2)
```



```
x <- seq(-pi,pi,len=100)
plot(x, sin(3*x/2+4), type='l')
```



```
curve(sin(3*x/2+4), from=-pi,to=pi)
```

