# TP2 - éléments de correction

automne 2021

## Exercice 1

### a)

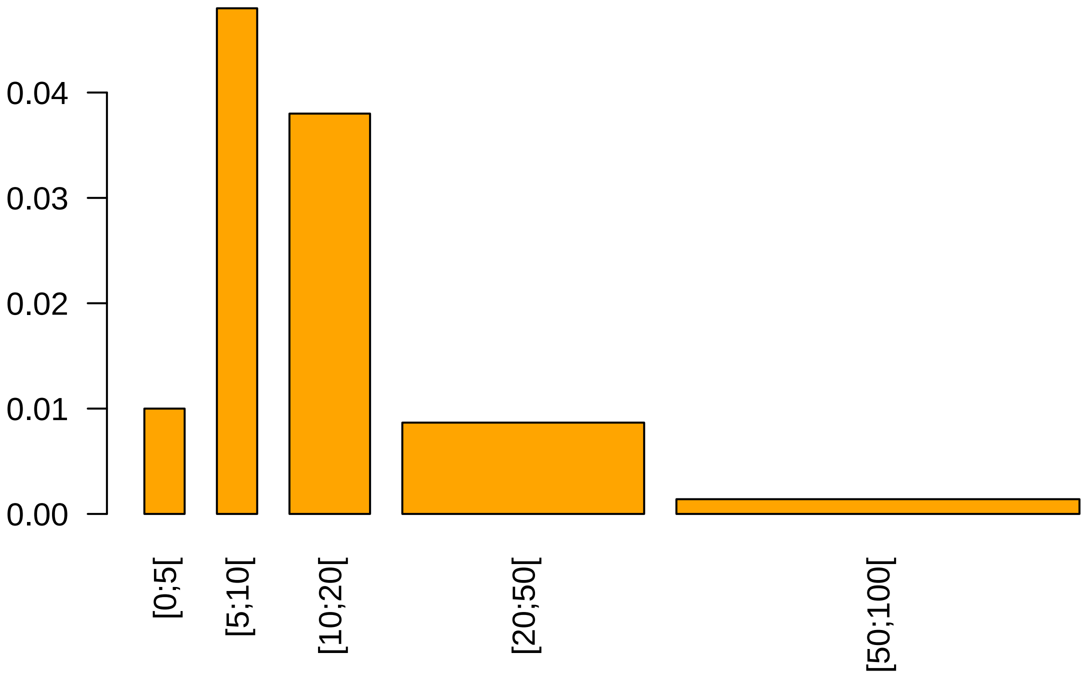
On souhaite construire un histogramme, donc un diagramme qui rend compte de la densité de la distribution des données. La densité $d_C$ associée à une classe $C = [a; b[$ est la proportion des individus $\pi_C = \frac{n_C}{N}$ appartenant à cette classe, divisée par l'amplitude de cette classe. On a donc :

$$d_C = \frac{n_C}{N} \frac{1}{b-a}.$$

Une façon de voir les choses est la suivante. On construit le diagramme en bâtons de façon à ce que la *surface* de chaque bâton soit égale à la proportion $\pi_C$ des individus appartenant à la classe correspondante $C$. La surface du bâton associé à la classe $C$ est $hauteur \times largeur = d_C \times (b-a) = \frac{n_C}{N} \frac{1}{b-a}(b-a) = \frac{n_C}{N} = \pi_C$.

```
superficie <- c("[0;5[","[5;10[","[10;20[","[20;50[","[50;100[")
bornes <- c(0, 5, 10, 20, 50, 100)
effectif <- c(5, 24, 38, 26, 7)
amplitudes <- diff(bornes)
centres <- bornes[-6] + amplitudes/2
hauteurs = effectif/sum(effectif)/amplitudes
```

```
barplot(height = hauteurs,
        width = amplitudes,
        names.arg = superficie,
        las = 2,
        col = 'orange')
```




### b)

Les effectifs et fréquences cumulés croissants et décroissants sont faciles à calculer.

```
ecc <- cumsum(effectif)
ecd <- rev(cumsum(rev(effectif)))
fcc <- cumsum(effectif)/sum(effectif)
fcd <- rev(cumsum(rev(effectif)))/sum(effectif)
```

### c)

Le mode est le centre de la classe de plus forte densité (la classe dont le bâton est le plus haut).

```
centres[which(rank(hauteurs)==5)]
```

```
## [1] 7.5
```

### d)

On reproduit artificiellement des données correspondant à la situation décrite dans l'énoncé en répétant les valeurs des centres le nombre de fois que cette classe est observée.

```
donnees <- rep(centres, effectif)
mean(donnees)
```

```
## [1] 21.975
```

```r
var(donnees)
```

```
## [1] 331.9817
```

### e)

Ce calcul est élémentaire, et peut se réaliser de plusieurs façons.

```r
median(donnees)
```

```
## [1] 15
```

```r
quantile(donnees, 0.5)
```

```
## 50%
##  15
```

## Exercice 2

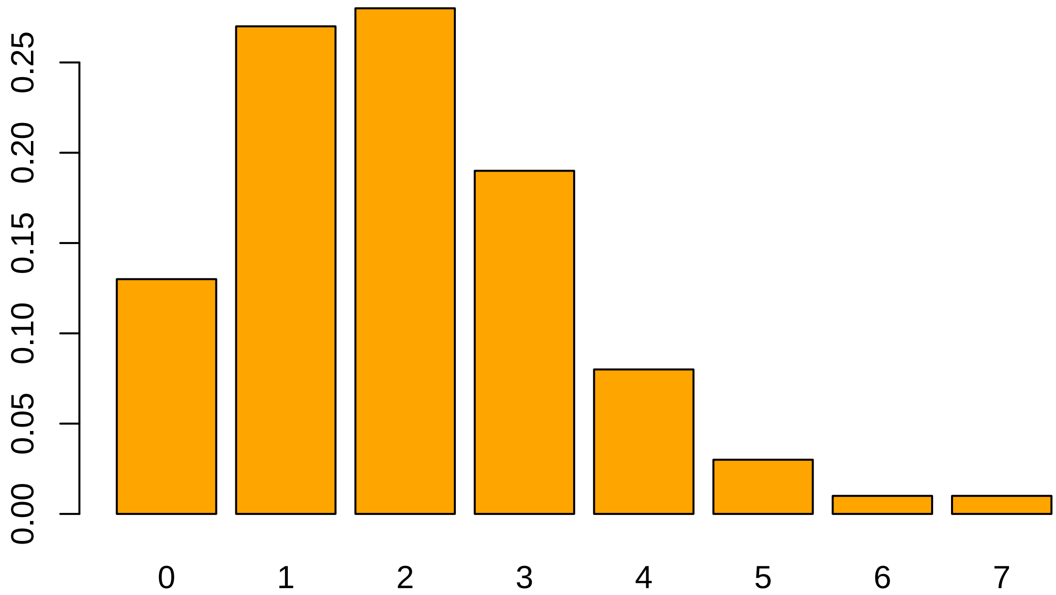
On commence par reporter les données de l'énoncé dans R.

```r
vers <- 0:7
prelevements <- c(13, 27, 28, 19, 8, 3, 1, 1)
```

### a)

L'unité statistique est le prélèvement. Une *observation* dans ce contexte est un échantillon de sable, dans lequel on compte le nombre de vers. L'unique variable ici est la variable quantitative discrète qu'est le nombre de vers. À chaque unité statistique (prélèvement), on associe la valeur de la variable "nombre de vers".

```r
barplot(height = prelevements/sum(prelevements),
        names.arg = vers,
        col = 'orange')
```




### b)

C'est facile.

```
fcc <- cumsum(prelevements)/sum(prelevements)
fcd <- rev(cumsum(rev(prelevements)))/sum(prelevements)
```

### c)

Comme dans l'exercice 1, on reproduit les données, cette fois-ci de façon exacte.

```
donnees <- rep(vers, prelevements)
mean(donnees)
```

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

```
var(donnees)
```

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

### d)

Ces calculs sont faciles. On utilise une petite astuce pour retrouver le mode.

```
vers[rank(prelevements)==8]
```

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

```r
quantile(donnees, 0.25)
```

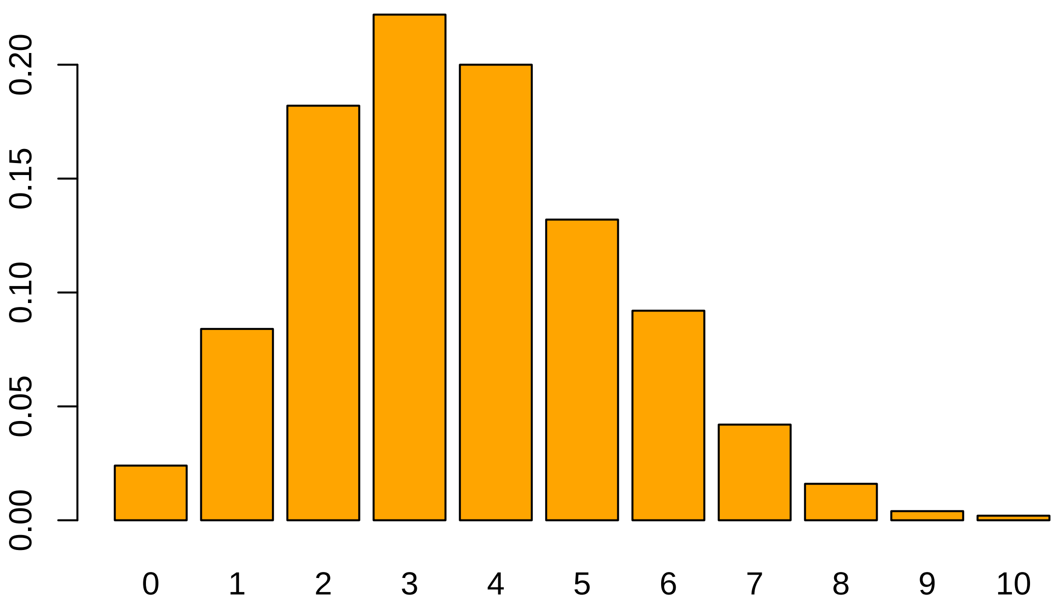
```
## 25%
##   1
```

# Exercice 3

## a)

L'unité statistique est le prélevement de sang. La variable est le décompte des hématies dans ce prélèvement ; c'est une cariable quantitative discrète à valeurs entières.

```r
i <- 0:10
ni <- as.integer(unlist(strsplit("12 42 91 111 100 66 46 21 8 2 1", split = " ")))
N <- 500
barplot(height = ni/N,
        names.arg = i,
        col = 'orange')
```




## b)

```r
freq <- ni/N
fcc <- cumsum(freq)
fcd <- rev(cumsum(rev(freq)))
tab <- as.table(rbind(freq, fcc, fcd))
colnames(tab) <- i
tab
```

```
##            0     1     2     3     4     5     6     7     8     9    10
```

```
## freq 0.024 0.084 0.182 0.222 0.200 0.132 0.092 0.042 0.016 0.004 0.002
## fcc  0.024 0.108 0.290 0.512 0.712 0.844 0.936 0.978 0.994 0.998 1.000
## fcd  1.000 0.976 0.892 0.710 0.488 0.288 0.156 0.064 0.022 0.006 0.002
```

### c,d)

Pour le mode :

```r
i[rank(ni)==max(rank(ni))]
```

```
## [1] 3
```

Pour les autres grandeurs, il y a deux méthodes.

#### 1)

```r
donnees <- rep(i, ni)
mu <- mean(donnees)
med <- median(donnees)
medbis <- quantile(donnees, 0.5)
sig2 <- var(donnees)
sig <- sd(donnees)
```

#### 2)

```r
mu <- sum(i*ni)/N
med <- i[fcc==min(fcc[fcc>=.5])]
sig2 <- sum(ni*(i-mu)**2)/(N-1)
sig <- sqrt(sig2)
```

## Exercice 4

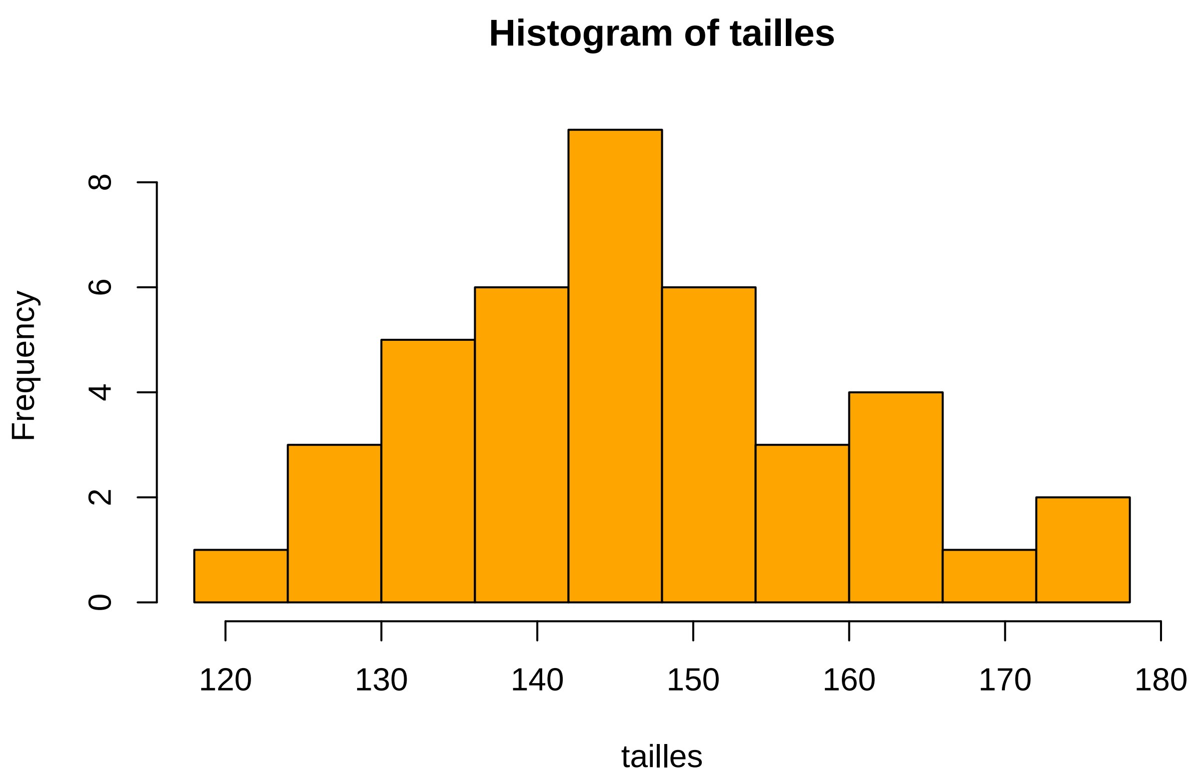
```r
tailles <- as.integer(unlist(strsplit("138 164 150 132 144 125 149 157 146 158 140 147 136 148 152 144
```

### a)

```r
mu <- mean(tailles)
sig2 <- var(tailles)
eid <- as.numeric(quantile(tailles, 0.9) - quantile(tailles, 0.1))
```

### b)

```
h <- hist(tailles,
     breaks = seq(118, 178, 6),
     col = 'orange')
```



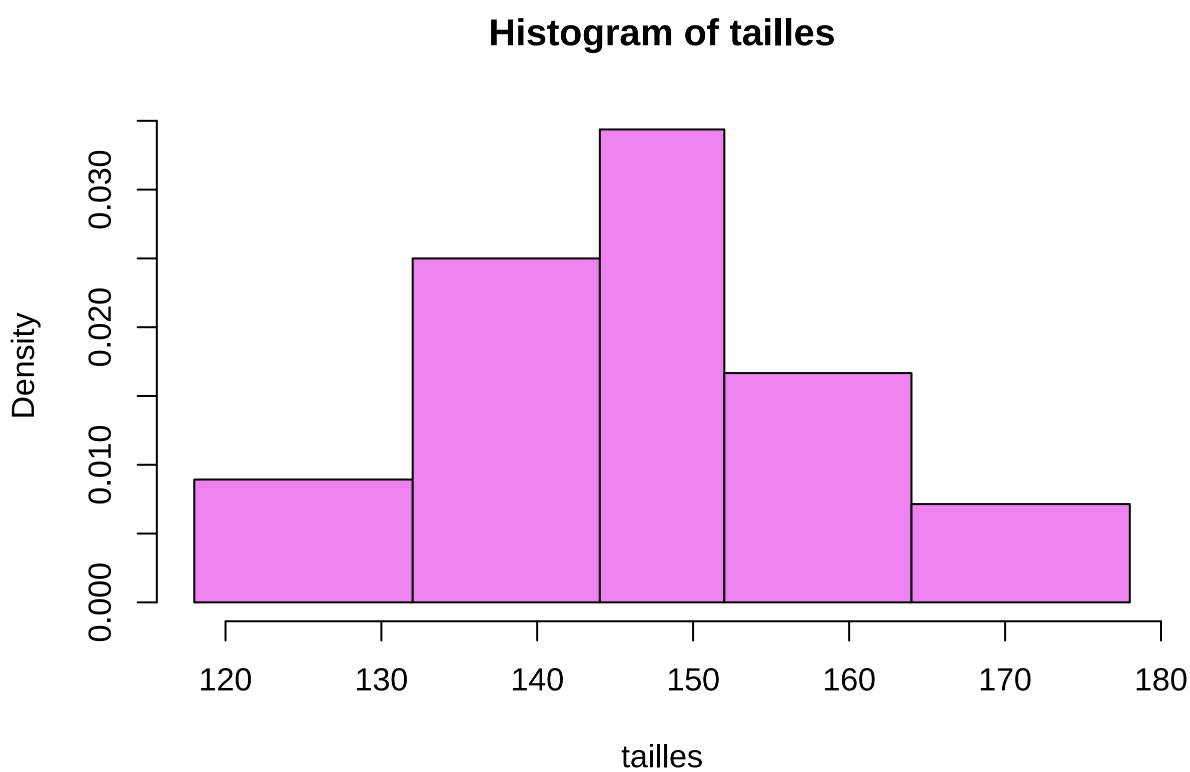

```
inter <- function(vec) {
  paste(c("[", vec[1], ";", vec[2], "["), collapse = "")
}


bornes <- matrix(rep(seq(118, 178, 6), c(1,rep(2,9),1)),
                 ncol = 2,
                 byrow = T)
classes <- apply(bornes, 1, inter)
tab <- as.table(rbind(h$counts))
colnames(tab) <- classes
rownames(tab) <- "effectif"
tab
```

```
##          [118;124[ [124;130[ [130;136[ [136;142[ [142;148[ [148;154[ [154;160[
## effectif         1         3         5         6         9         6         3
##          [160;166[ [166;172[ [172;178[
## effectif         4         1         2
```

```
centres <- apply(bornes, 1, mean)
mu <- sum(centres*h$counts)/length(tailles)
sig2 <- sum(h$counts*(centres-mu)**2)/(length(tailles)-1)
```

**c)**

```
h <- hist(tailles,
     breaks = c(118, 132, 144, 152, 164, 178),
     col = 'violet')
```




```
bornes <- matrix(rep(c(118, 132, 144, 152, 164, 178), c(1,rep(2,4),1)),
                 ncol = 2,
                 byrow = T)
classes <- apply(bornes, 1, inter)
tab <- as.table(rbind(h$counts))
colnames(tab) <- classes
rownames(tab) <- "effectif"
tab
```

```
##          [118;132[ [132;144[ [144;152[ [152;164[ [164;178[
## effectif         5        12        11         8         4
```

```
centres <- apply(bornes, 1, mean)
mu <- sum(centres*h$counts)/40
sig2 <- sum(h$counts*(centres-mu)**2)/39
mu
```

```
## [1] 146.425
```

```r
sig2
```

```
## [1] 170.8147
```