Travaux Dirigés ISV51 - Statistiques descriptives - Correction

Julien Chiquet

23 octobre et 6 novembre 2015

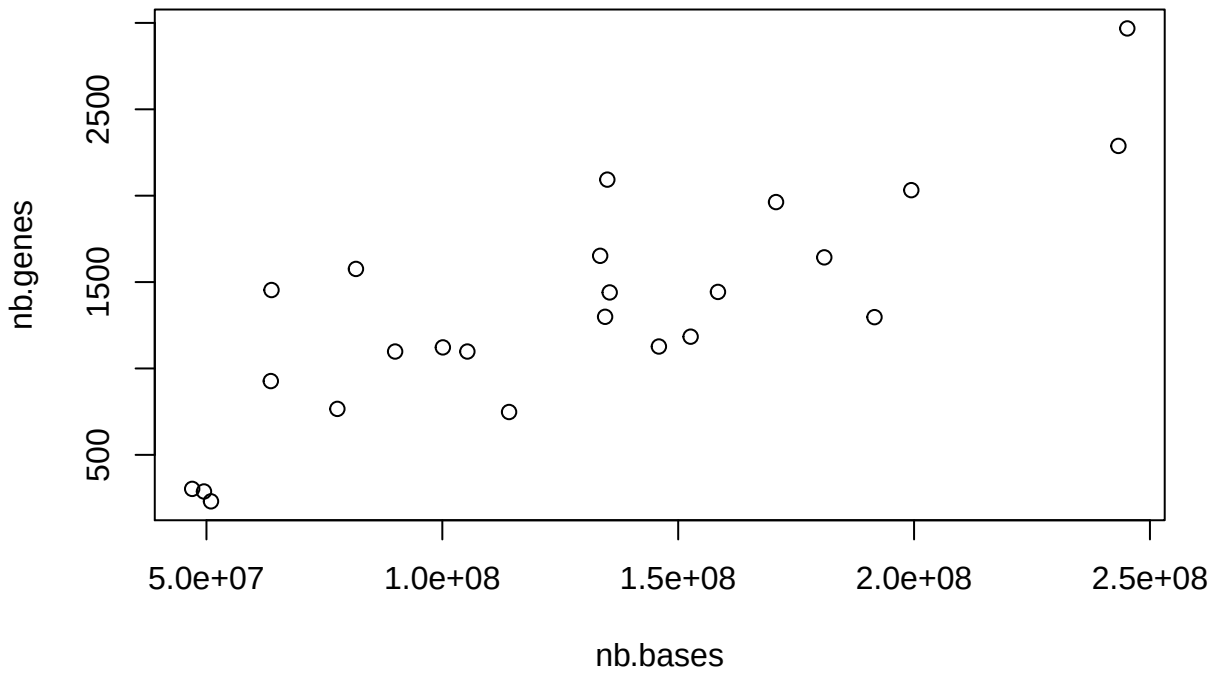
Exercice 1: lecture/écriture d'un tableau de données - chromosomes

1. Créer un tableau à 24 lignes et 3 colonnes en lisant le fichier chromosomes.txt avec la fonction `read.table`. Chaque ligne représentera un chromosome humain (22 autosome, 2 chromosomes sexuels) et les colonnes seront respectivement leur noms, nombre de gènes, et longueur en bases.

```
chromosomes <- read.table("chromosomes.txt", col.names=c("chr","nb.genes","nb.bases"))
```

2. Représenter Le nombre de gène en fonction du nombre de bases.

```
plot(nb.genes~nb.bases, chromosomes)
```



3. Ajouter une colonne supplémentaire au tableau qui spécifie pour chaque chromosome s'il est autosome ou pas.

```
chromosomes$autosome <- c(rep(TRUE,22),FALSE,FALSE)
```

4. Calculer le nombre total de gènes et de paires de base d'un génome humain (pour un homme, puis une femme).

```
commun <- colSums(subset(chromosomes, autosome==TRUE, c("nb.genes", "nb.bases")))
homme <- commun + colSums(subset(chromosomes, chr == "X" | chr == "Y", c("nb.genes", "nb.bases")))
femme <- commun + 2 * subset(chromosomes, chr == "X", c("nb.genes", "nb.bases"))
```

Exercice 2: lecture données, graphiques - somnifère

Pour étudier l'effet d'un somnifère, on mesure chez 20 patients le nombre d'heures de sommeil supplémentaires par rapport à la durée moyenne de leur nuit sans traitement. On obtient les résultats suivants:

# patient	extra
1	0.7
2	-1.6
3	-0.2
4	-1.2
5	-0.1
6	3.4
7	3.7
8	0.8
9	0.0
10	2.0
11	1.9
12	0.8
13	1.1
14	0.1
15	-0.1
16	4.4
17	5.5
18	1.6
19	4.6
20	3.4

1. Saisir ces données dans un vecteur.

```
somnifere <- scan(text="0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0.0 2.0 1.9 0.8 1.1 0.1 -0.1 4.4 5.5 1.6 4.
```

2. Faites un résumé numérique.

```
summary(somnifere)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.600  -0.025   0.950   1.540   3.400   5.500
```

```
fivenum(somnifere)
```

```
## [1] -1.60 -0.05  0.95  3.40  5.50
```

3. Tracer un diagramme en tige et feuille.

```r
stem(somnifere)
```

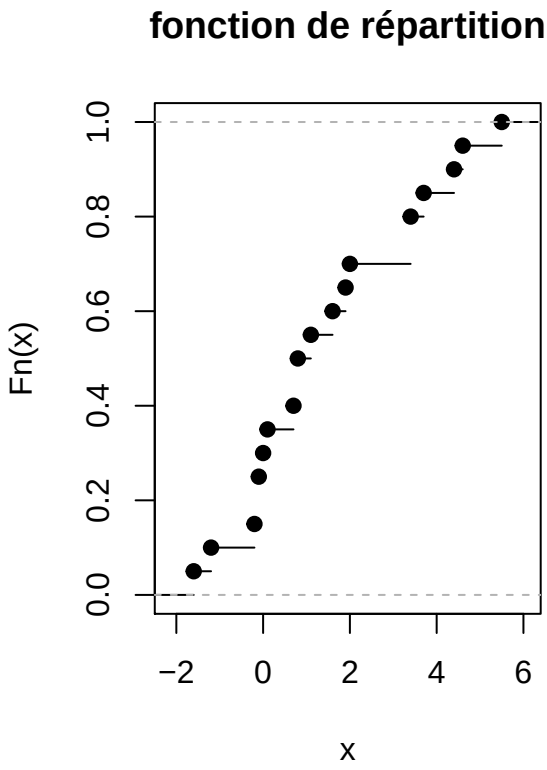
```
##
##   The decimal point is at the |
##
##   -0 | 62211
##    0 | 01788169
##    2 | 0447
##    4 | 465
```

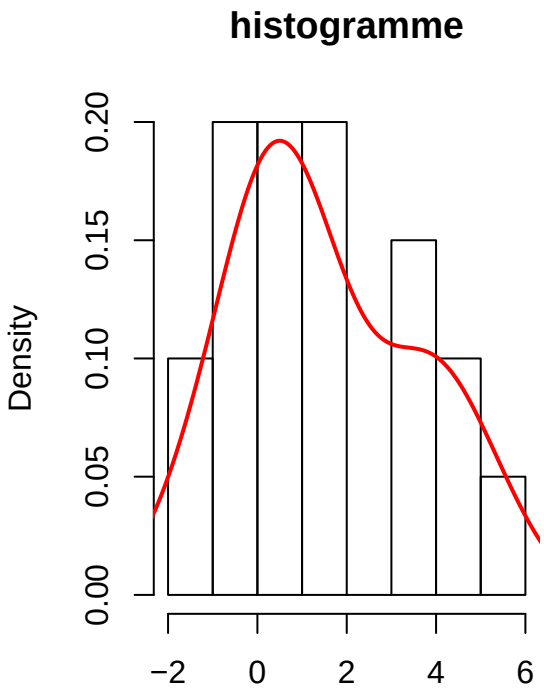
```r
stem(somnifere, scale = 2)
```

```
##
##   The decimal point is at the |
##
##   -1 | 62
##   -0 | 211
##    0 | 01788
##    1 | 169
##    2 | 0
##    3 | 447
##    4 | 46
##    5 | 5
```

4. Tracer la fonction de répartition empirique puis l'histogramme normalisé des données dans la même fenêtre graphique.

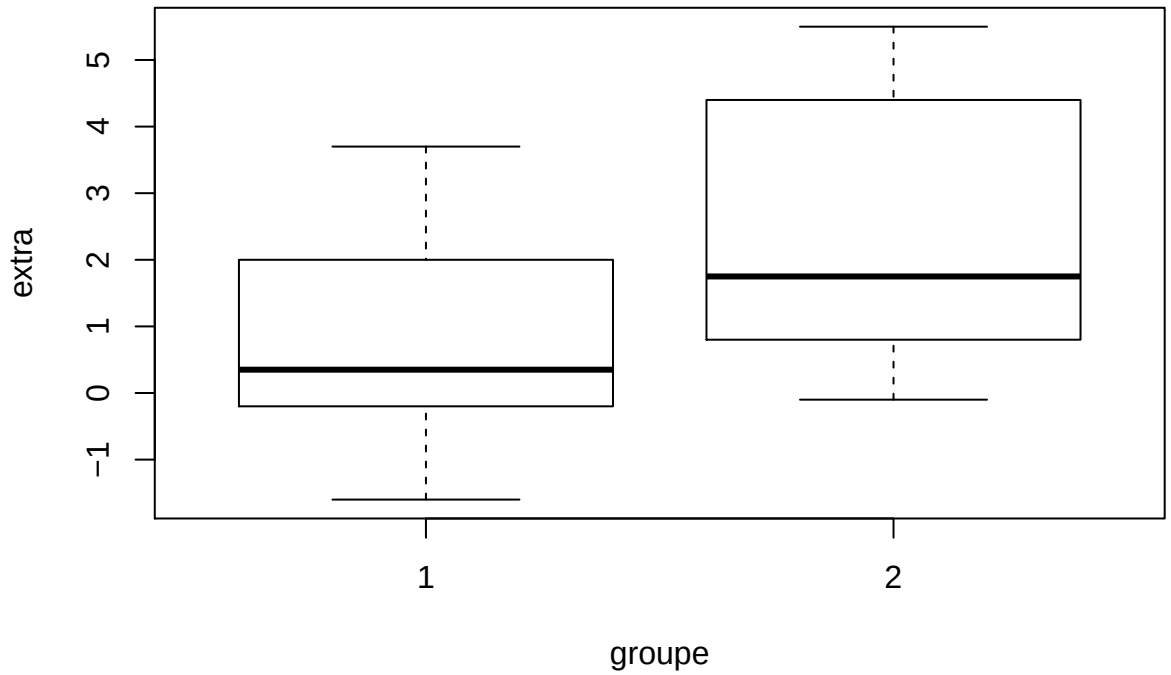
```r
par(mfrow=c(1,2))
plot(ecdf(somnifere), main="fonction de répartition")
hist(somnifere, freq=FALSE, main="histogramme", xlab="")
lines(density(somnifere), col="red", lwd=2)
```





5. Ces données sont en fait issues de deux groupes d'individus: apposer une variable indiquant le groupe associé à l'observation de la variable `extra` sachant que les 10 premiers individus sont issus du groupe 1 et les 10 suivants du groupe 2 (utiliser, par exemple, la commande `data.frame`). Faire un résumé statistique pour chaque groupe et tracer alors les boîtes à moustaches des observations selon les groupes. Qu'en pensez-vous ?

```
somnifere <- data.frame(extra=somnifere, groupe=factor(rep(c(1,2), each=10)))
plot(extra ~ groupe, somnifere)
```



Il semble y avoir un "effet groupe" : le somnifère est plus efficace dans le second groupe. Une analyse de la variance à un facteur permettrait de répondre à cette question.

Exercice 3: variables qualitative - variations génétiques dans les populations humaines.

1. Charger le jeu de données `hdpg` du package `ade4` et lire son descriptif.

```
library(ade4)
data(hdpg)
```

2. Nous considérerons le tableau `hdpg$ind` qui décrit l'échantillon des 1066 individus de l'étude.

```
ind <- hdpg$ind; rm(hdpg)
head(ind)
```

```
##   id sex population region
## 1  1   1     Brahui   Asia
## 2  3   1     Brahui   Asia
## 3  5   1     Brahui   Asia
## 4  7   1     Brahui   Asia
## 5  9   1     Brahui   Asia
## 6 11   1     Brahui   Asia
```

3. Combien de populations différentes participent à l'étude ?

```
nlevels(ind$population)
```

```
## [1] 52
```

4. Dresser les tableaux des effectifs des variable population, région et sexe.

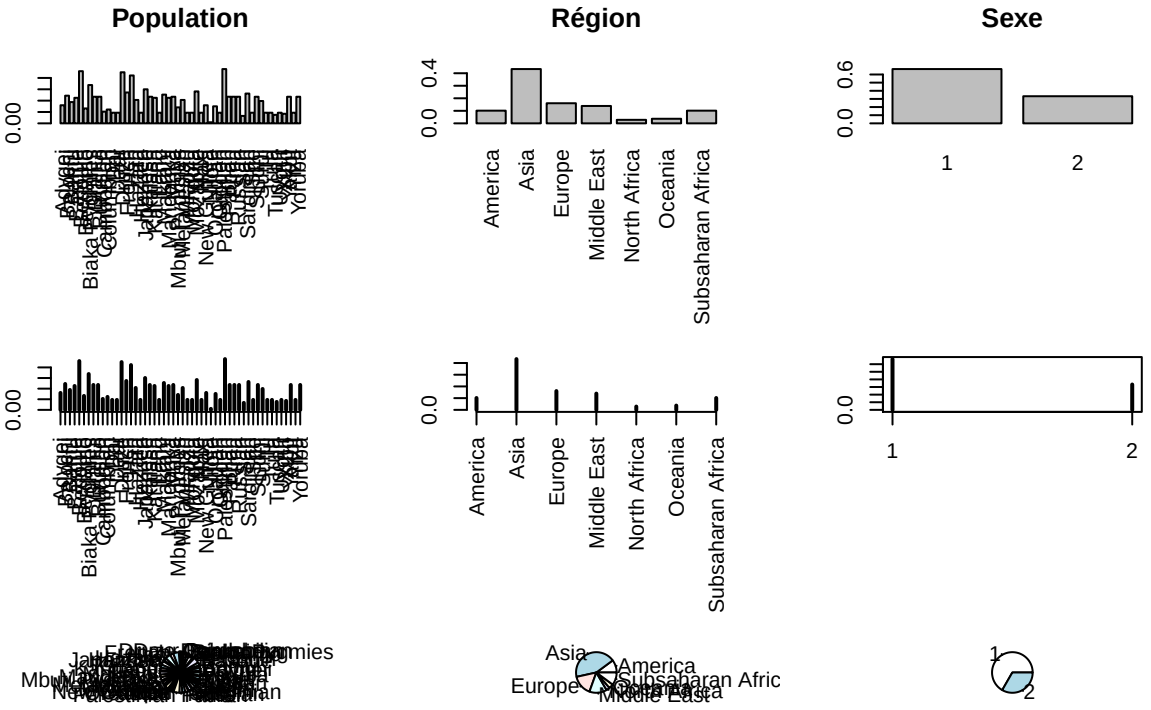
```
eff.pop <- table(ind$population)
eff.reg <- table(ind$region)
eff.sex <- table(ind$sex)
```

5. Transformer ces tableaux en tableaux de fréquences.

```
freq.pop <- eff.pop/sum(eff.pop)
freq.reg <- eff.reg/sum(eff.reg)
freq.sex <- eff.sex/sum(eff.sex)
```

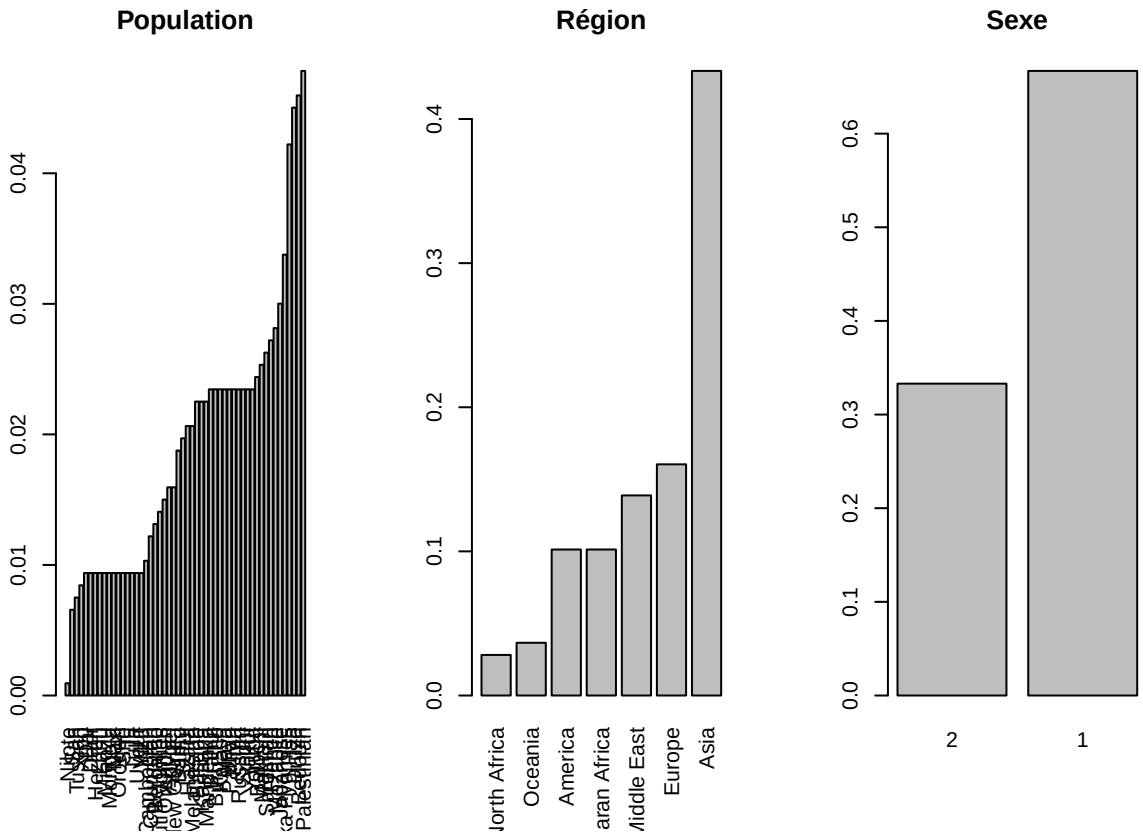
6. Représenter vos tableaux de fréquence par des diagramme en bâton, et par des camemberts.

```
par(mfrow=c(3,3))
barplot(freq.pop, las=3, main="Population")
barplot(freq.reg, las=3, main="Région")
barplot(freq.sex, main="Sexe")
plot(freq.pop, type="h", las=3, xlab="", ylab="")
plot(freq.reg, type="h", las=3, xlab="", ylab="")
plot(freq.sex, type="h", xlab="", ylab="")
pie(freq.pop); pie(freq.reg) ; pie(freq.sex)
```



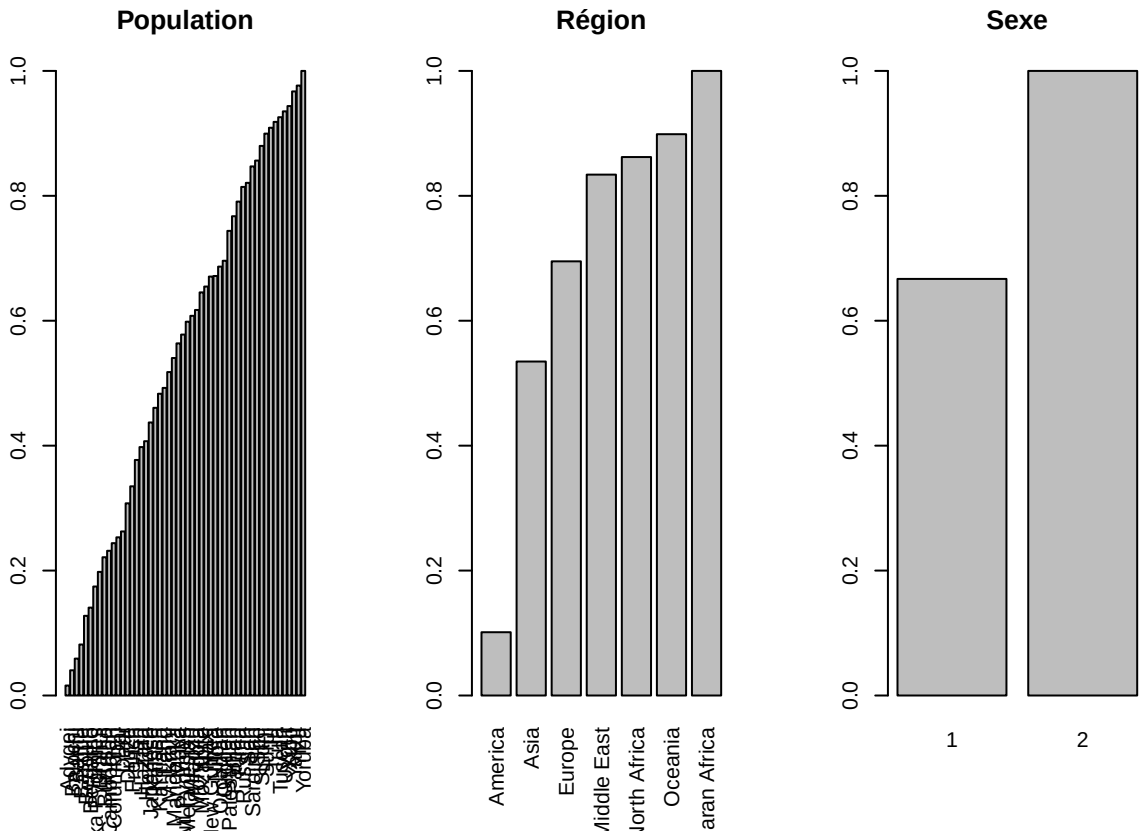
On aurait pu ordonné les fréquences pour clarifier un peu la lecture.

```
par(mfrow=c(1,3))
barplot(sort(freq.pop), las=3, main="Population")
barplot(sort(freq.reg), las=3, main="Région")
barplot(sort(freq.sex), main="Sexe")
```



7. Représenter les fréquences cumulées.

```
par(mfrow=c(1,3))
barplot(cumsum(freq.pop), las=3, main="Population")
barplot(cumsum(freq.reg), las=3, main="Région")
barplot(cumsum(freq.sex), main="Sexe")
```



8. Commenter les représentations.

Exercice 4: variable quantitative - coefficient de Gini

Le coefficient de Gini permet de mesurer l'inégalité des revenus dans une population. Si tous les individus gagnent le même salaire le coefficient de Gini vaut 0 (situation égalitaire), alors que si un seul individu gagne tous le revenu disponible et les autres rien l'index de gini vaut 1. Les états-unis ont par exemple un coefficient de Gini de 0.47.

1. Charger le jeu de données gini.Rdata.

```
load("gini.RData")
head(gini)
```

```
##   year country     gini
## 1 2004 Austria 25.77157
## 2 2005 Austria 26.12778
## 3 2006 Austria 25.33235
## 4 2007 Austria 26.14727
## 5 2004 Belgium 27.00748
## 6 2005 Belgium 28.53019
```

2. Sélectionner les lignes du tableau correspondant à l'année 2007.

```r
gini2007 <- subset(gini, year == 2007, -year)
```

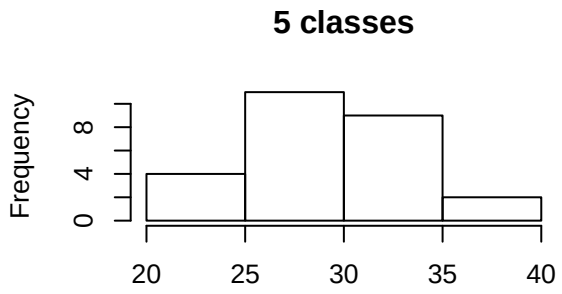
3. Tracer un diagramme en tige et feuille des coefficients.

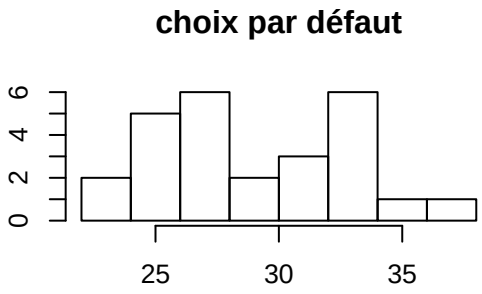
```r
stem(gini2007$gini, scale=2)
```

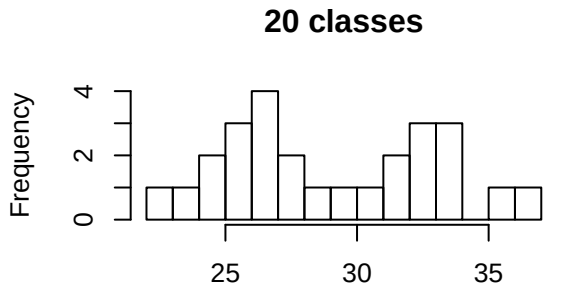
```
##
##   The decimal point is at the |
##
##   22 | 9
##   23 | 4
##   24 | 22
##   25 | 227
##   26 | 1234
##   27 | 46
##   28 | 1
##   29 | 8
##   30 | 6
##   31 | 25
##   32 | 228
##   33 | 48
##   34 | 0
##   35 | 7
##   36 | 9
```

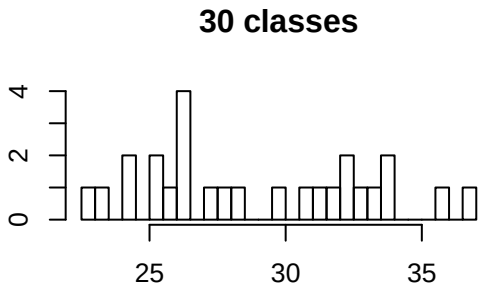
4. Tracer l'histogramme des coefficients.

```r
par(mfrow=c(2,2))
hist(gini2007$gini, nclass = 5, main="5 classes", xlab="")
hist(gini2007$gini, main="choix par défaut", xlab="", ylab="")
hist(gini2007$gini, nclass = 20, main="20 classes", xlab="")
hist(gini2007$gini, nclass = 30, main="30 classes", xlab="", ylab="")
```



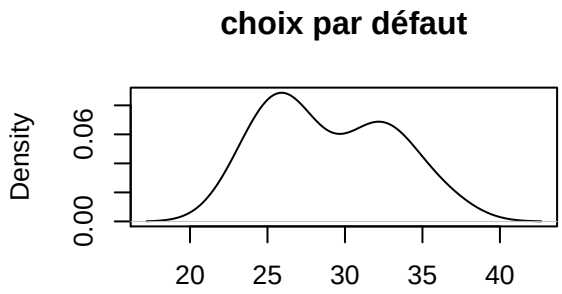


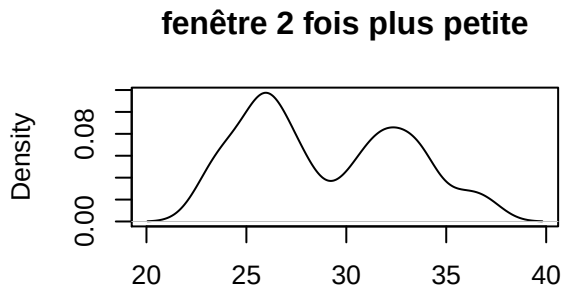


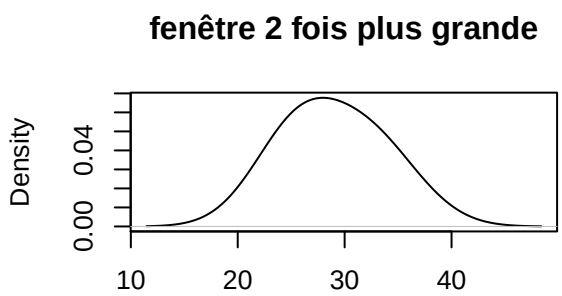


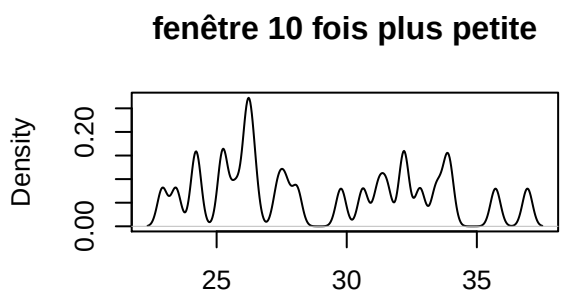
5. Tracer l'histogramme lissé des coefficients.

```
par(mfrow=c(2,2))
plot(density(gini2007$gini), main="choix par défaut", xlab="")
plot(density(gini2007$gini, adjust=.5),main="fenêtre 2 fois plus petite", xlab="")
plot(density(gini2007$gini, adjust=2) , main="fenêtre 2 fois plus grande", xlab="")
plot(density(gini2007$gini , adjust=.1), main="fenêtre 10 fois plus petite", xlab="")
```



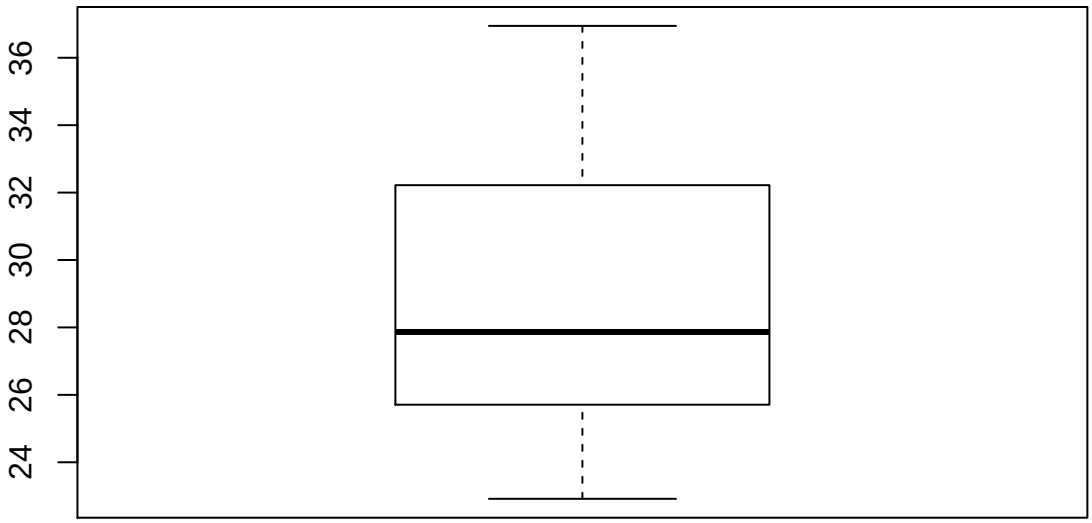






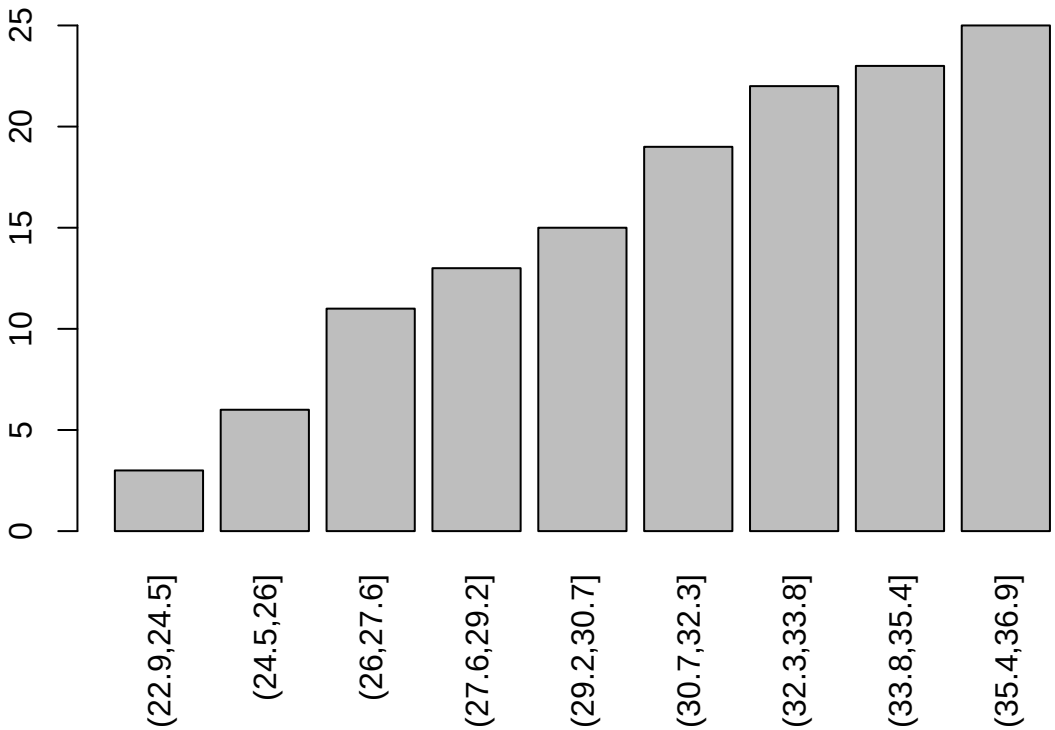
6. Tracer le boxplot des coefficients.

```
boxplot(gini2007$gini)
```



7. Tracer un diagramme des fréquences cumulées des coefficients.

```
bornes <- seq(min(gini2007$gini), max(gini2007$gini), len=10)
freq.gini <- table(cut(gini2007$gini, bornes))
barplot(cumsum(freq.gini), las=3)
```



8. Écrire une fonction R qui rende les pays de coefficient Gini d'index maximum et minimum.

```r
extremeGini <- function(ginicoef, country) {

  return(c( min.gini = country[which.min(ginicoef)],
            max.gini = country[which.max(ginicoef)]))

}
extremeGini(gini2007$gini, gini2007$country)
```

```
##   min.gini   max.gini
## "Slovenia" "Portugal"
```

9. Classer les pays par leur coefficient de Gini.

```r
with(gini2007, country[order(gini)])
```

```
##  [1] "Slovenia"       "Sweden"         "Slovakia"       "Norway"
##  [5] "Czech.Republic" "Denmark"        "Hungary"        "Austria"
##  [9] "Finland"        "Belgium"        "France"         "Luxembourg"
## [13] "Netherlands"    "Slovenia"       "Cyprus"         "Germany"
## [17] "Ireland"        "Spain"          "Poland"         "Italy"
## [21] "United.Kingdom" "Estonia"        "Lithuania"      "GR"
## [25] "Latvia"         "Portugal"
```

10. Calculer la moyenne, la variance, le coefficient d'asymétrie, le coefficient d'aplatissement pour la distribution des coefficients de gini. Commenter.

```r
library(moments)
mean(gini2007$gini); var(gini2007$gini); skewness(gini2007$gini); kurtosis(gini2007$gini)
```

```
## [1] 28.97361
```

```
## [1] 16.55891
```

```
## [1] 0.2794213
```

```
## [1] 1.856416
```

11. Combien de pays sont plus égalitaires que la France en europe.

```r
gini.France <- with(gini2007, gini[which(country == "France")])
sum(gini2007$gini > gini.France)
```

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

Exercice 5: simulation et graphiques - puces à ADN

Les distributions des intensités moyennes des spots d'ADNc correspondants aux gènes exprimés et non exprimés peuvent être modélisées par deux gaussiennes. On suppose que la première distribution a une espérance $\mu^{\text{e}} = 1000$ et un écart type $\sigma^{\text{e}} = 100$; la seconde distribution a pour espérance $\mu^{\text{ne}} = 400$ et pour écart type $\sigma^{\text{ne}} = 150$.

Chaque gène correspond à 4 spots répliqués. L'expression d'un gène est définie comme la moyenne des 4 spots qui lui sont associés.

Analyse élémentaire du modèle

1. Créer sous `R` les variables `mu.e`, `sigma.e`, `mu.ne` et `sigma.ne` et affectez-y les valeurs de l'énoncé.

```
mu.e <- 1000; mu.ne <- 400; sigma.e <- 100; sigma.ne <- 150
```

2. On note S^{e} la variable aléatoire décrivant l'intensité d'un spot correspondant à un gène exprimé. Quelle est la probabilité pour que S^{e} ait une valeur inférieure ou égale à 700 ?

Un simple calcul montre que $\mathbb{P}(S^{\mathrm{e}} \leq 700) = \Phi((700 - \mu^{\mathrm{e}})/\sigma^{\mathrm{e}})$. On peut vérifier que les tables de la loi normale de `R` et sous forme papier sont d'accord entre elles:

```
pnorm(700, mu.e, sigma.e)
```

```
## [1] 0.001349898
```

3. On note G^{e} la variable aléatoire décrivant le niveau d'expression d'un gène exprimé. Quelle est la probabilité pour que G^{e} ait une expression inférieure ou égale à 700 ?

On exprime le niveau d'expression d'un gène comme la somme de uatre spots, donc $G^{\mathrm{e}} = 1/4 \sum_{i=1}^{4} S_i^{\mathrm{e}}$. La probabilité recherchée est bien évidemment $\mathbb{P}(G^{\mathrm{e}} \leq 700) = \Phi((700 - \mu^{\mathrm{e}})/(\sigma^{\mathrm{e}}/\sqrt{4}))$, égale à

```
pnorm(700, mu.e, sigma.e/2)
```

```
## [1] 9.865876e-10
```

4. On introduit la variable aléatoire G^{ne} pour les gènes non exprimés. Quelle est la valeur seuil t telle que la probabilité d'avoir G^{e} inférieure ou égale à t soit égale à la probabilité d'avoir G^{ne} supérieure à t ?

On cherche la valeur t vérifiant $\mathbb{P}(G^{\mathrm{e}} \leq t) = \mathbb{P}(G^{\mathrm{ne}} > t)$, soit,

$$\mathbb{P}(G^{\mathrm{e}} \leq t) = 1 - \mathbb{P}(G^{\mathrm{ne}} \leq t) \Leftrightarrow \Phi\left(\frac{t - \mu^{\mathrm{e}}}{\sigma^{\mathrm{e}}/2}\right) = \Phi\left(-\frac{t - \mu^{\mathrm{ne}}}{\sigma^{\mathrm{ne}}/\sqrt{4}}\right) \Leftrightarrow \frac{t - \mu^{\mathrm{e}}}{\sigma^{\mathrm{e}}/2} = -\frac{t - \mu^{\mathrm{ne}}}{\sigma^{\mathrm{ne}}/2},$$

d'où $t = 760$

5. Quelle est la probabilité d'avoir un gène exprimé dont l'expression est inférieur à t (faux négatif) ? La probabilité d'un faux négatif est $\mathbb{P}(G^{\mathrm{e}} \leq t)$, soit

```
t <- 760
pnorm((t-mu.e)/(sigma.e/2))
```

```
## [1] 7.933282e-07
```

6. Quelle est la probabilité d'avoir un gène non exprimé dont l'expression est supérieure à t (faux positif) ?

La probabilité d'un faux positif est $\mathbb{P}(G^{\mathrm{ne}} \geq t)$, soit

```
t <- 760
1-pnorm((t-mu.ne)/(sigma.ne/2))
```

```
## [1] 7.933282e-07
```

Simulations et graphiques

1. Générer $n = 1000$ intensités de spots correspondant aux gènes exprimés et non exprimés. Les stocker dans les vecteurs `spots.e` et `spots.ne`. Parmi tous les spots générés, stocker la plus petite et la plus grande valeur observée dans des variables `MIN` et `MAX`.

Commençons par simuler les populations. Il est utile de repérer l'amplitude des données générées pour pouvoir définir les bornes des axes du graphe par la suite. On stocke donc `MIN` et le `MAX` afin de calibrer les axes lors des sorties graphiques.

```
n <- 1000
spots.e  <- rnorm(n,mu.e,sigma.e)
spots.ne <- rnorm(n,mu.ne,sigma.ne)
MIN <- min(spots.e,spots.ne)
MAX <- max(spots.e,spots.ne)
```

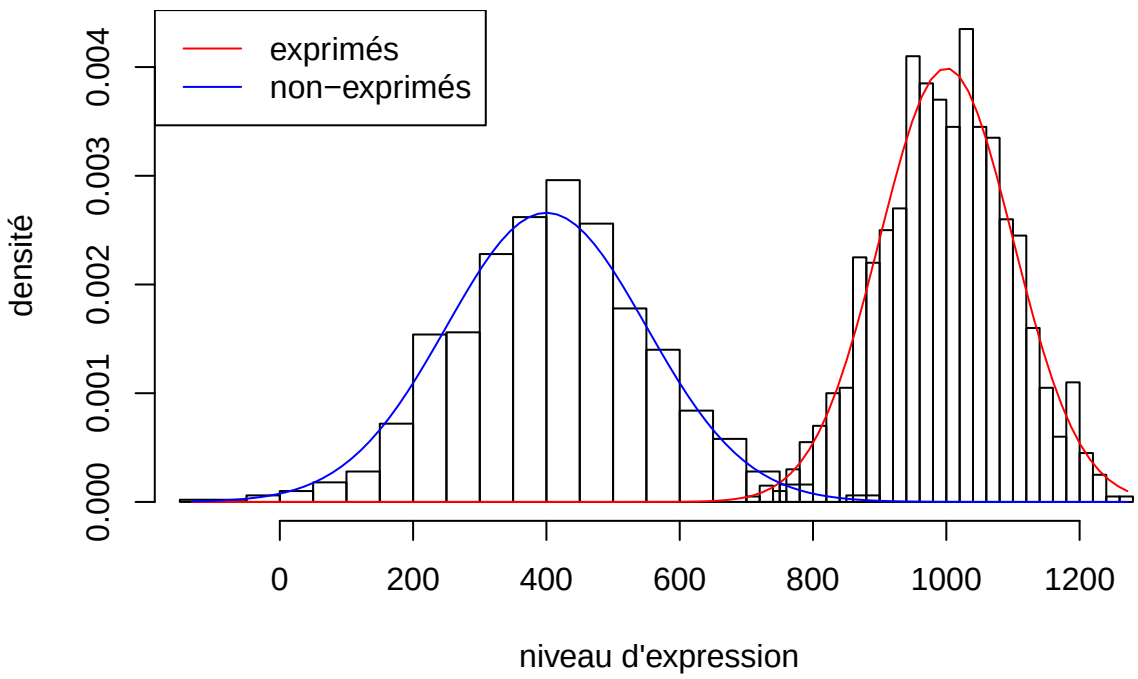
2. Créer deux objets de classe histogramme, sans les tracer, correspondant à chacune des deux populations de spots et stocker les dans des variables `hist.e` et `hist.ne`.

On stocke les objets histogrammes afin de pouvoir les faire apparaître dans la même fenêtre graphique.

```
hist.e  <- hist(spots.e ,nclass=30,plot=FALSE)
hist.ne <- hist(spots.ne,nclass=30,plot=FALSE)
```

3. Tracer sur un même graphique les deux histogrammes normalisés et les densités théoriques (fonction `curve`). Utiliser deux couleurs différentes pour les deux populations de spots. Apposer une légende au graphe (commande `legend`).

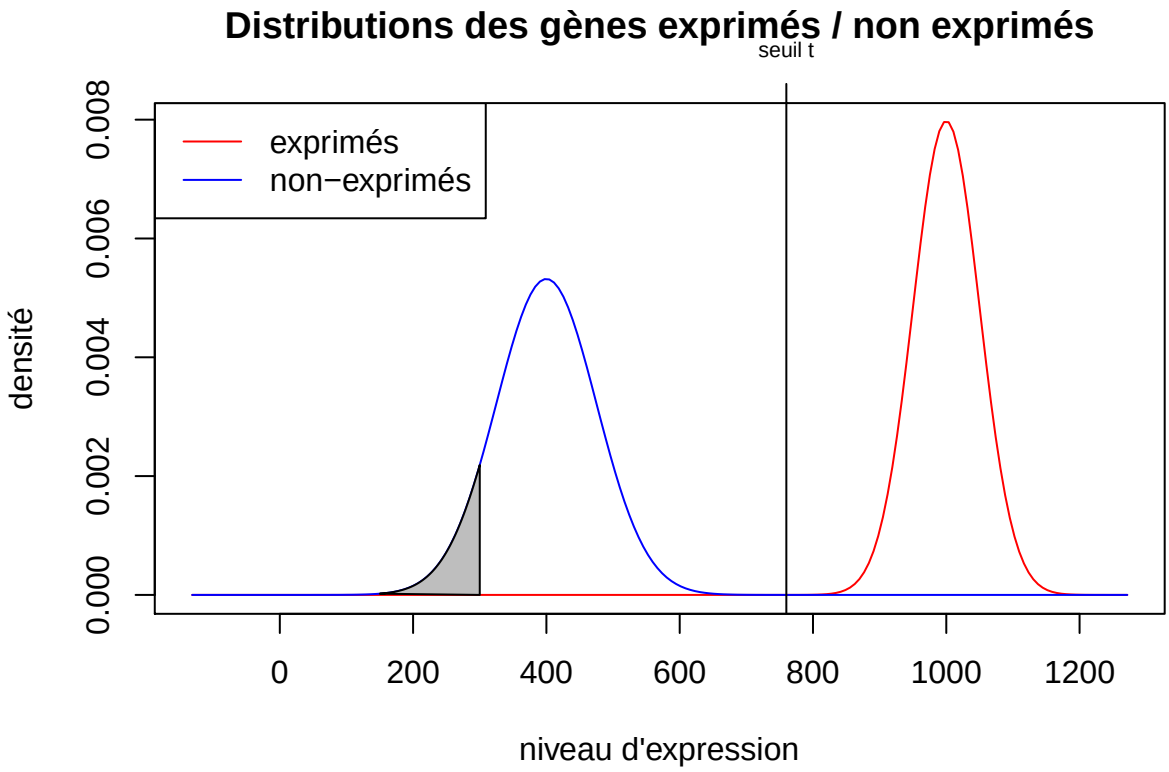
```
title <- "Distributions des spots exprimés / non exprimés"
plot(hist.e,freq=FALSE,xlim=c(MIN,MAX),main=title,
     xlab="niveau d'expression", ylab="densité")
plot(hist.ne,freq=FALSE,add=TRUE)
curve(dnorm(x,mu.e,sigma.e)  ,from=MIN,to=MAX,
      add=TRUE,col="red")
curve(dnorm(x,mu.ne,sigma.ne),from=MIN,to=MAX,
      add=TRUE,col="blue")
legend("topleft",c("exprimés","non-exprimés"),
       col=c("red","blue"),lty=c(1,1))
```



4. Tracer sur un même graphique les densités théoriques des gènes exprimés et non exprimés. Faire une légende. Puis, à l'aide de la commande `polygon`, représenter l'aire sous courbe correspondant à la probabilité pour qu'un gène non exprimé ait une expression inférieure à 300. Enfin, tracer une droite verticale indiquant l'emplacement du seuil t (commande `abline`).

De prime abord, la fonction polygone peut sembler compliquer à utiliser. En fait, il n'en est rien! Il suffit de penser à rajouter un point d'ordonnée $y = 0$ car le polygone est tracé en joignant le premier au dernier point. Prenez le temps de regarder ce que ça donne lorsque l'on oublie ce point!

```
title <- "Distributions des gènes exprimés / non exprimés"
curve(dnorm(x,mu.e,sigma.e/2), n=200,from=MIN,
      to=MAX,col="red", main=title, ylab="densité",
      xlab="niveau d'expression")
curve(dnorm(x,mu.ne,sigma.ne/2), n=200,from=MIN,
      to=MAX,add=TRUE,col="blue")
legend("topleft",c("exprimés","non-exprimés"),
       col=c("red","blue"),lty=c(1,1))
x <- seq(150,300,len=100)
y <- dnorm(x,mu.ne,sigma.ne/2)
x <- c(x,300)
y <- c(y,0)
polygon(x,y,col="gray")
abline(v=t)
axis(3, at=t, labels="seuil t", cex.axis=0.7)
```


Distributions des gènes exprimés / non exprimés
seuil t
exprimés
non−exprimés
densité
0.000
0.002
0.004
0.006
0.008
0
200
400
600
800
1000
1200
niveau d'expression