Expression Data Analysis

```
library(opera)
```

```
## Warning: package 'opera' was built under R version 3.6.2
```

Exercice 1: Rappels de Statistiques

Question 1

On cherche la probabilité qu'un spot correspondant à un gène exprimé possède une valeur inférieure ou égale à 700.

```
pnorm(700,1000,100)
```

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

Question 2

Quelle est la probabilité qu'un gène exprimé possède une expression inférieur ou égale à 700 ? Pour les details des calculs voir le fichier pdf svp.

```
pnorm(700,1000,100/sqrt(4))
```

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

Question 3

Quel est la valeur seuil t telle que la probablité d'avoir l'expression d'un gène exprimé inf´erieure ou égale à t soit égale à la probabilité d'avoir l'expression d'un gène non exprimé supérieur à t. Voir pd svp. On obtient $t^* = 760$

Question 4

Quelle est la probabilité d'avoir un gène exprimé dont l'expression est inférieur à t (faux négatif)?

```
pnorm(760,1000,100/sqrt(4))
```

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

Question 5

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

```
1-pnorm(760,400,150/sqrt(4))
```

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

Question 6

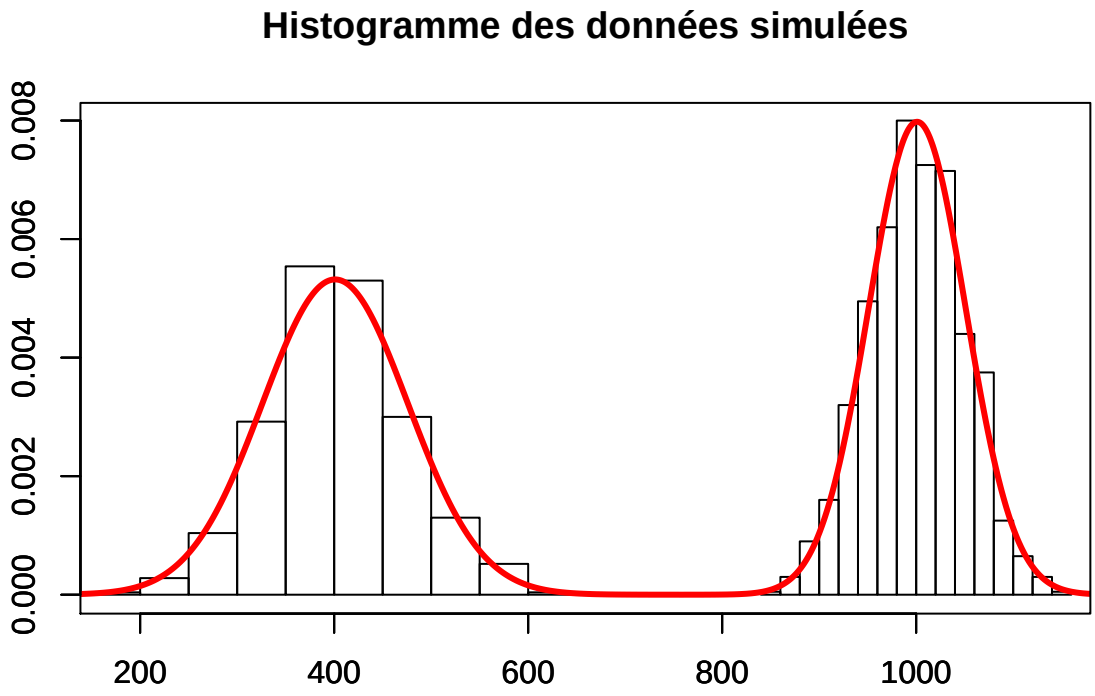
Simulation des données

```
x1 = apply(matrix(rnorm(4*1000, 1000, 100),nrow = 4),2,mean)
x2 <- apply(matrix(rnorm(4*1000, 400, 150),nrow = 4),2,mean)
```

Histogrammes

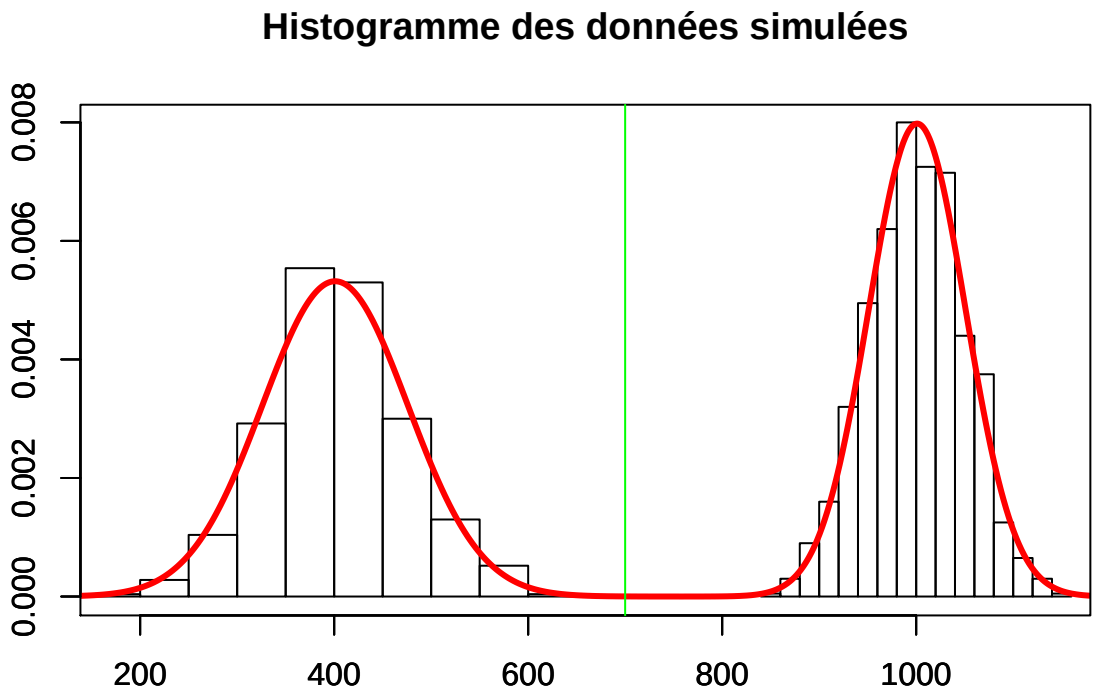
```
xlim <- c(floor(min(x1,x2)),floor(max(x1,x2))+1)
ylim <- c(0,dnorm(1000,1000,100/sqrt(4)))
d = apply(rbind(dnorm(c(0:2000),400,150/sqrt(4)),dnorm(c(0:2000),1000,100/sqrt(4))),2,max) #la densité

hist(x1,xlim = xlim,freq = F, ylim = ylim, xlab = "", ylab = "", main = "")
par(new = T)
hist(x2,xlim = xlim,freq = F, ylim = ylim, xlab = "", ylab = "", main = "")
par(new = T)
plot(d, ylim = ylim, xlim = xlim, type = "l", xlab = "", ylab = "", main =
       "Histogramme des données simulées", lwd = 3, col = "red")
```




```
hist(x1,xlim = xlim,freq = F, ylim = ylim, xlab = "", ylab = "", main = "")
par(new = T)
hist(x2,xlim = xlim,freq = F, ylim = ylim, xlab = "", ylab = "", main = "")
par(new = T)
```

```
plot(d, ylim = ylim, xlim = xlim, type = "l", xlab = "", ylab = "", main =
       "Histogramme des données simulées", lwd = 3, col = "red")
abline(v = 700, col = "green")
```




```
length(which(x1<=700))
```

```
## [1] 0
```

Exercice 2: Test d'hypothèses

Importation des données

```
library(rda)
data(colon)
```

```
dim(colon.x)
```

```
## [1]   62 2000
```

```
length(colon.y)
```

```
## [1] 62
```

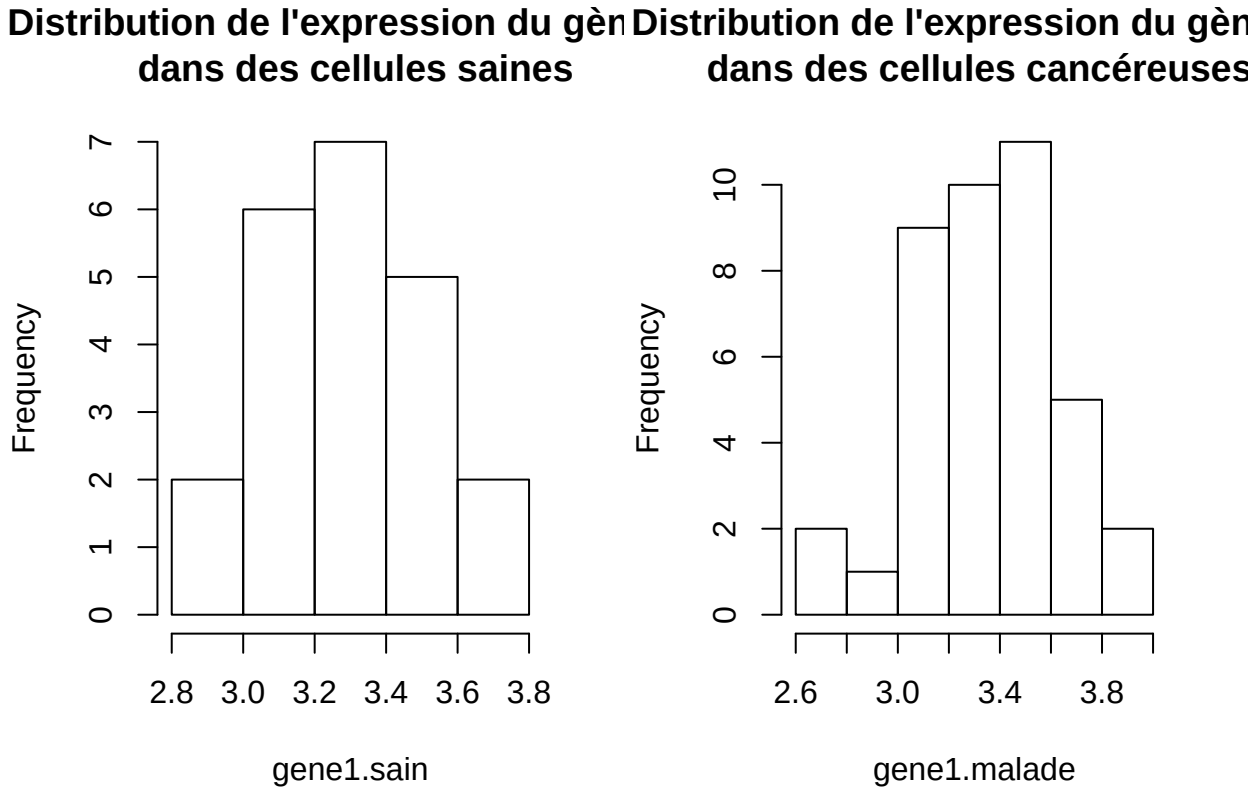
```
colon = data.frame(colon.x, colon.y)
```

Question 1

```
gene1.sain = colon[colon$colon.y == 1 ,1]
gene1.malade = colon[colon$colon.y == 2 ,1]
```

Partie a

```
par(mfrow = c(1,2))
hist(gene1.sain, main = "Distribution de l'expression du gène 1 \n dans des cellules saines")
hist(gene1.malade, main = "Distribution de l'expression du gène 1 \n dans des cellules cancéreuses")
```



Ave un teste de Student on vérifie que comme nous l'indiquent les histogrammes ci dessus, il n'y a pas de difference d'expression significative du gène 1 dans des cellules sines ou cancéreuses.

```
test1 = t.test(gene1.sain, gene1.malade, var.equal = T)
test1
```

```
##
##  Two Sample t-test
##
## data:  gene1.sain and gene1.malade
## t = -0.80087, df = 60, p-value = 0.4264
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.19819065  0.08486261
```

```
## sample estimates:
## mean of x mean of y
##  3.293923  3.350587
```

Partie b

```
test1$p.value
```

```
## [1] 0.426365
```

On ne peut donc pas rejeter l'hypothèse nulle c'est à dire l'égalité d'expression du gène 1 dans les 2 conditions décrites (cellules saines VS cellules cancéreuses).

Partie c

On cherche maintenant les 10 gènes dont l'expression dans les deux condition est la plus significativement différente.

```
# On fait tous les test et on stock les pvaleurs dans un vecteur p
p= c()
for(i in 1:2000){
  p[i] = t.test(colon[,i], colon[,2001], var.equal = T)$p.value
}
#On ordonne le vecteur p dans l'ordre croissant et on selectionne les 10 plus petites valeures
p2 = order(p, decreasing = F)[1:10]
```

```
#On affiche les 10 gènes dont l'expression dans les deux condition est la plus significativement différ
colnames(colon)[p2]
```

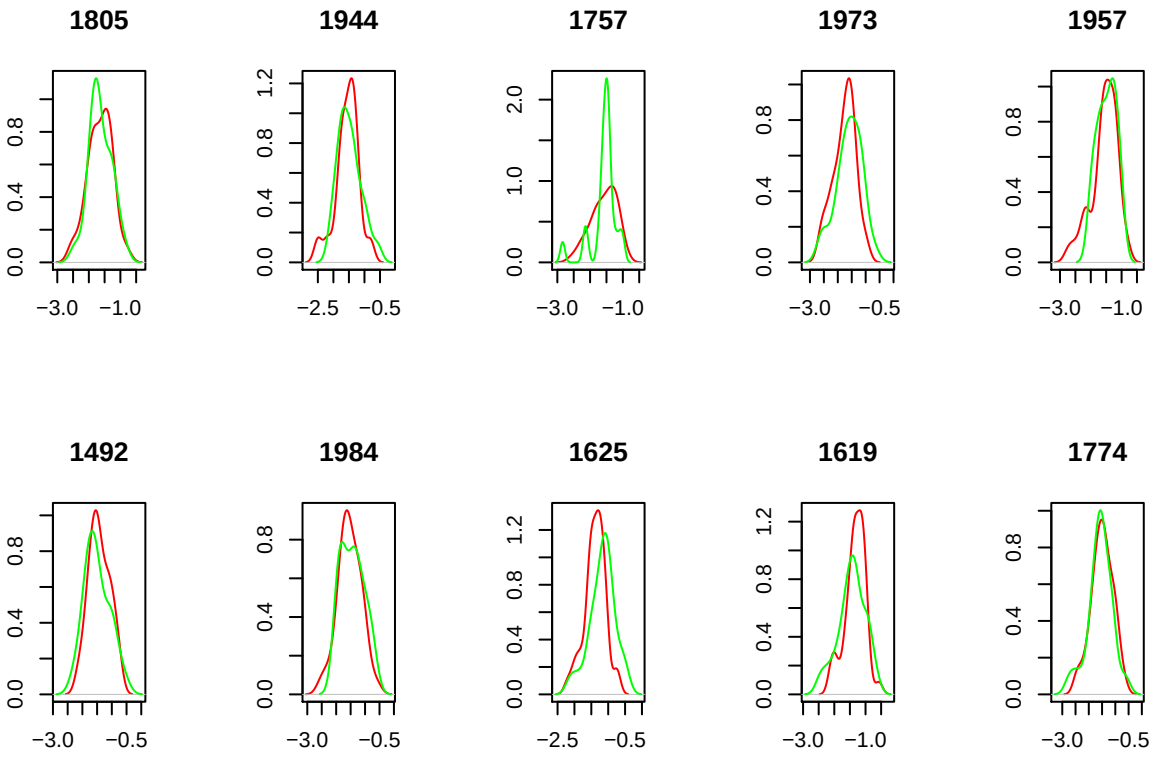
```
##  [1] "X1805" "X1944" "X1757" "X1973" "X1957" "X1492" "X1984" "X1625"
##  [9] "X1619" "X1774"
```

```
par(mfrow = c(2,5))
plots_m = c()
plots_s = c()
for(i in 1:length(p2)){
  m =  colon[colon.y==2,p2[i]]
  s =  colon.x[colon.y==1,p2[i]]

  densi_m <- density(m)
  densi_s <- density(s)

  xlim = c(min(densi_m$x,densi_s$x),max(densi_m$x,densi_s$x))
  ylim = c(0,max(densi_m$y,densi_s$y))

  plot(densi_m, xlim = xlim, ylim = ylim, xlab = "", ylab = "", main =  p2[i], col = "red")
  lines(densi_s, xlim = xlim, ylim = ylim, xlab = "", ylab = "", main = p2[i], col = "green")
#par(new = T)
#plot(densi_sain, xlim = xlim, ylim = ylim, xlab = "", ylab = "", main = "", col = "blue")
}
```



On voit que les ditributions de l'expression de ces 10 gènes dans les cellules saines (vert) sont vraiment différentes que dans les cellules cancéreuses (rouge).

Question 2

Simulation des données

```
n = 1000
simu = matrix(nrow = n, ncol = 20)
for(i in 1:n){
simu[i,]=c(rnorm(n = 10, mean = 0, sd = 1), rnorm(n = 10, mean = 2, sd = 1))}
```

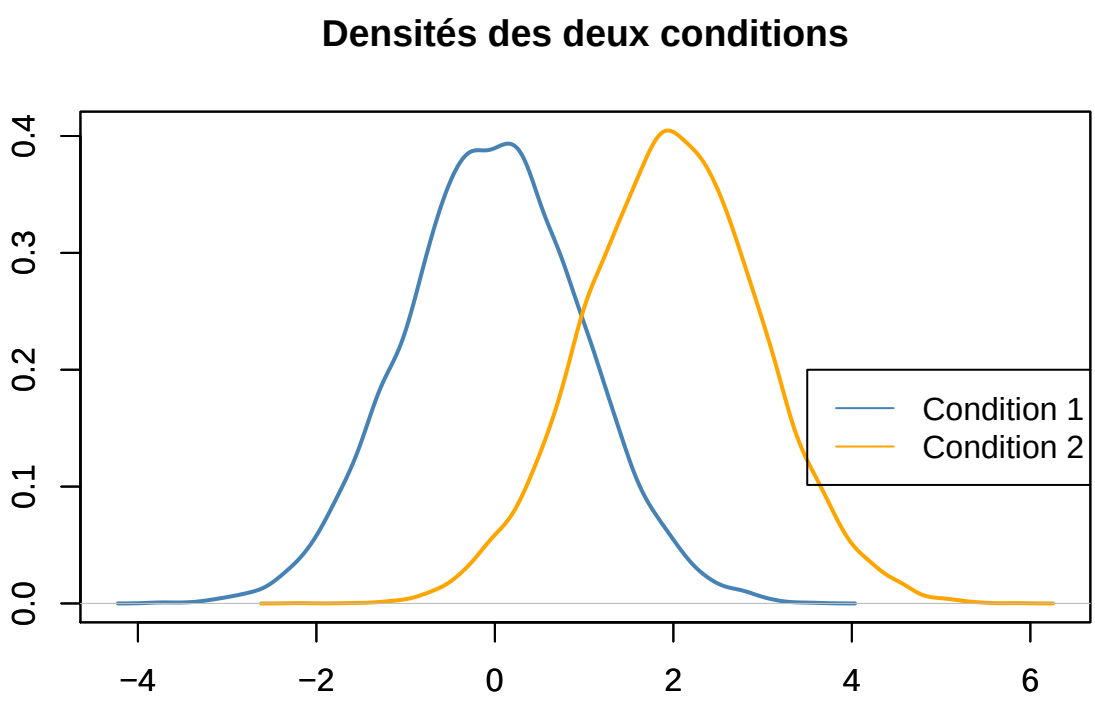
```
cond1.expr = simu[,1:10]
cond2.expr = simu[,11:20]
```

```
d_cond1 = density(cond1.expr)
d_cond2 = density(cond2.expr)


xlim1 = c(min(d_cond1$x,d_cond2$x),max(d_cond1$x,d_cond2$x))
ylim1 = c(0,max(d_cond1$y,d_cond2$y))


plot(d_cond1, xlim = xlim1, ylim = ylim1, xlab = "", ylab = "", main = "Densités des deux conditions",
par(new = T)
plot(d_cond2, xlim = xlim1, ylim1 = ylim, xlab = "", ylab = "", main = "", col = "orange", lwd = 2)
```

```
## Warning in plot.window(...): "ylim1" n'est pas un paramètre graphique
## Warning in plot.xy(xy, type, ...): "ylim1" n'est pas un paramètre graphique
## Warning in axis(side = side, at = at, labels = labels, ...): "ylim1" n'est
## pas un paramètre graphique
## Warning in axis(side = side, at = at, labels = labels, ...): "ylim1" n'est
## pas un paramètre graphique
## Warning in box(...): "ylim1" n'est pas un paramètre graphique
## Warning in title(...): "ylim1" n'est pas un paramètre graphique
legend(col = c("steelblue", "orange"),legend = c("Condition 1", "Condition 2"), x = 3.5, y= .2,lty = 1)
```



Partie a: Estimation des faux positifs avec les données simulées

Pour un gne i fixé

$$(H_0) : m^i_{cond1} = m^i_{cond2} \; et \; (H_1) : m^i_{cond1} \neq m^i_{cond2}$$

De plus, on fait l'hypothèse que tous les gènes sont exprimés differement dans les deux condition. Ainsi tous les gènes dont les expressions dans les deux conditions ne sont pas significativement differents sont des faux négatifs.

```
signif = 0
pval = c()
for(i in 1:n){
  pval[i] = t.test(simu[i,1:10], simu[i,11:20], var.equal = T)$p.value
  if(pval[i] <= 0.05) signif = signif +1}
signif
```

```
## [1] 983
```

```
FN = n -signif
FN
```

```
## [1] 17
```

On obtient donc 11 faux négatifs (soit 1.1%). ### Partie b

Question 3 Méthodes de Bonferroni et Sidak

```
BS = p.adjust(pval,method = "bonferroni")
#BS
cat(sum(pval <= 0.05), sum(BS <= 0.05))
```

```
## 983 301
```

Ainsi avec le correction de Bonferroni Sidak nous n'obtenons beaucoup moins de faux négatifs: on sait qu'il s'agit d'une amélioration puisque le nombre de vrai négatifs est 0 (hypotèse que chaque gène s'exprime differement dans les deux conditions).

Question 4 : Méthodes de Bonferroni et Hocheberg

```
BH <- p.adjust(pval,method = "BH")
cat(sum(pval <= 0.05), sum(BH <= 0.05))
```

```
## 983 983
```

Cette correction ne differe que très peu d'un test de student usuel.

Question 5

Ansi la méthode de Bonferoni Hocheberg va dans le sens de l'exercice c'est à dire qu'on rejette preque aussi souvent l'hypothèse nulle: c'est l'assurance qu'on la rejette à raison (d'ailleurs on le sait puisque les gènes sont supposés tous avoir une expression différente dans les deux conditions).