Projet Analyse de données

Mame Diarra Toure- Imane ALLA

12/24/2019

Exercice 1: Simulation d'un mélange à trois composantes.

On considère un mélange à 3 composantes de Bernoulli.
Dans cet exercice, nous avons décidé de reprogrammer notre propre simulation afin de mieux comprendre le processus, nous avons par la suite comparé notre simulation avec la simulation donnée.

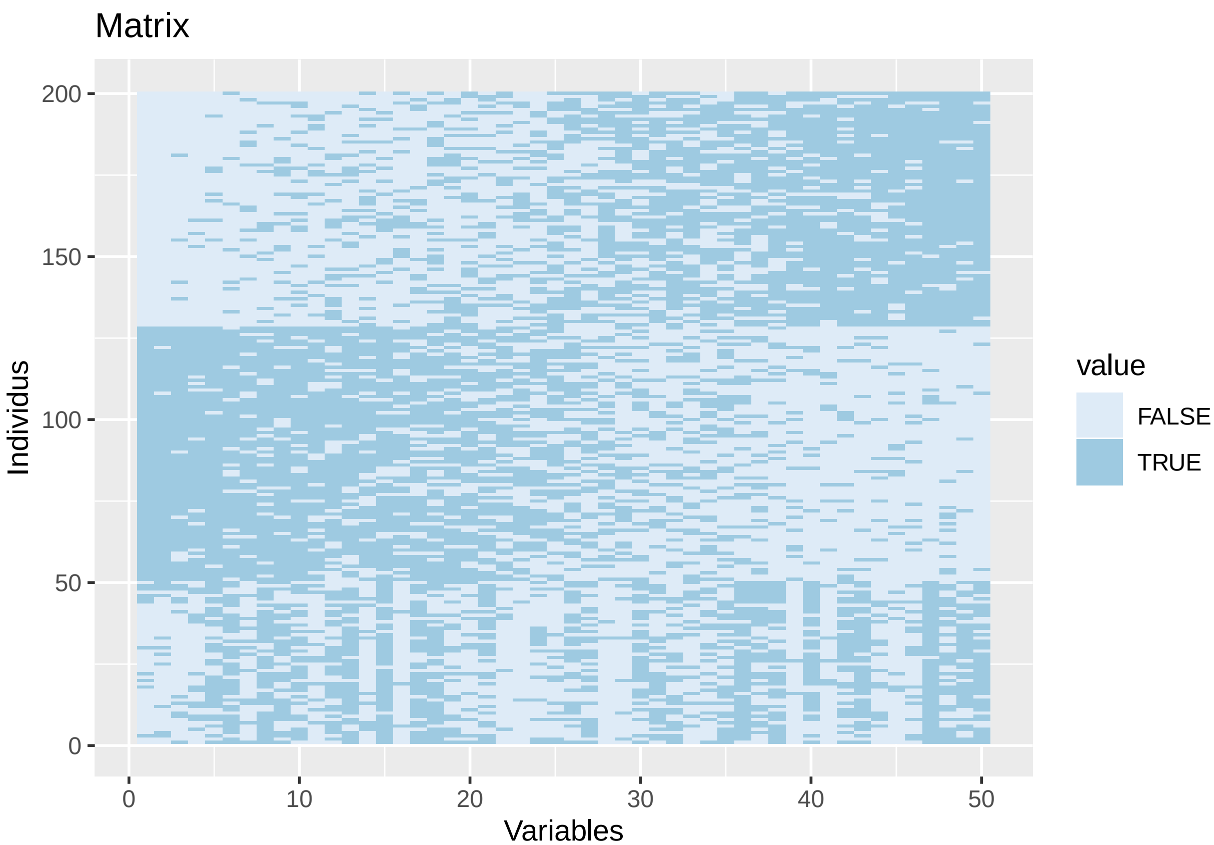
```
set.seed(3)
K<-3
p<-50
n<-200
pi<-c(1/3,1/3,1/3)
M<-matrix(runif(K*p),K,p)
M[K,]<-1-M[1,]
nks<-rmultinom(1,200,prob = pi)
Z<-rep(1:length(nks),nks)
X <-do.call(rbind,
                  mapply(function(nk,k){
                    matrix(rbernoulli(nk*p,p=M[k,]),
                           nrow = nk,
                           ncol=p,
                           byrow = TRUE)}, nks,1:K))

kmeans(X,3,nstart = 10)->res.kmeans
tidyData<-melt(X[order(res.kmeans$cluster),order(M[1,])])
```

```
set.seed(3)
K<-3
p<-50
n<-200
pi<-c(1/3,1/3,1/3)
#chaque ligne correspond a une composante de la mixture
M<-matrix(runif(K*p),K,p)
#Pour éviter d'avoir des valeurs similaires
M[K,]<-1-M[1,]
#Subdivision de l'échantillon: Combien appartiennent à chacune des lois(1 ère,2ème...)
nks<-rmultinom(1,200,prob = pi)
Z<-rep(1:length(nks),nks)
#la loi de X sachant Z chaque ligne correspond a un vecteur Xi
XsZ <- matrix(345,50,200-nks[3])
for (i in 1:nks[1]){
  XsZ[,i] <- rbernoulli(50,M[1,])
}
for(i in 1:(nks[2])){
  XsZ[,nks[1]+i] <- rbernoulli(50,M[2,])
}

XsZ3 <- matrix(345,50,nks[3])
```

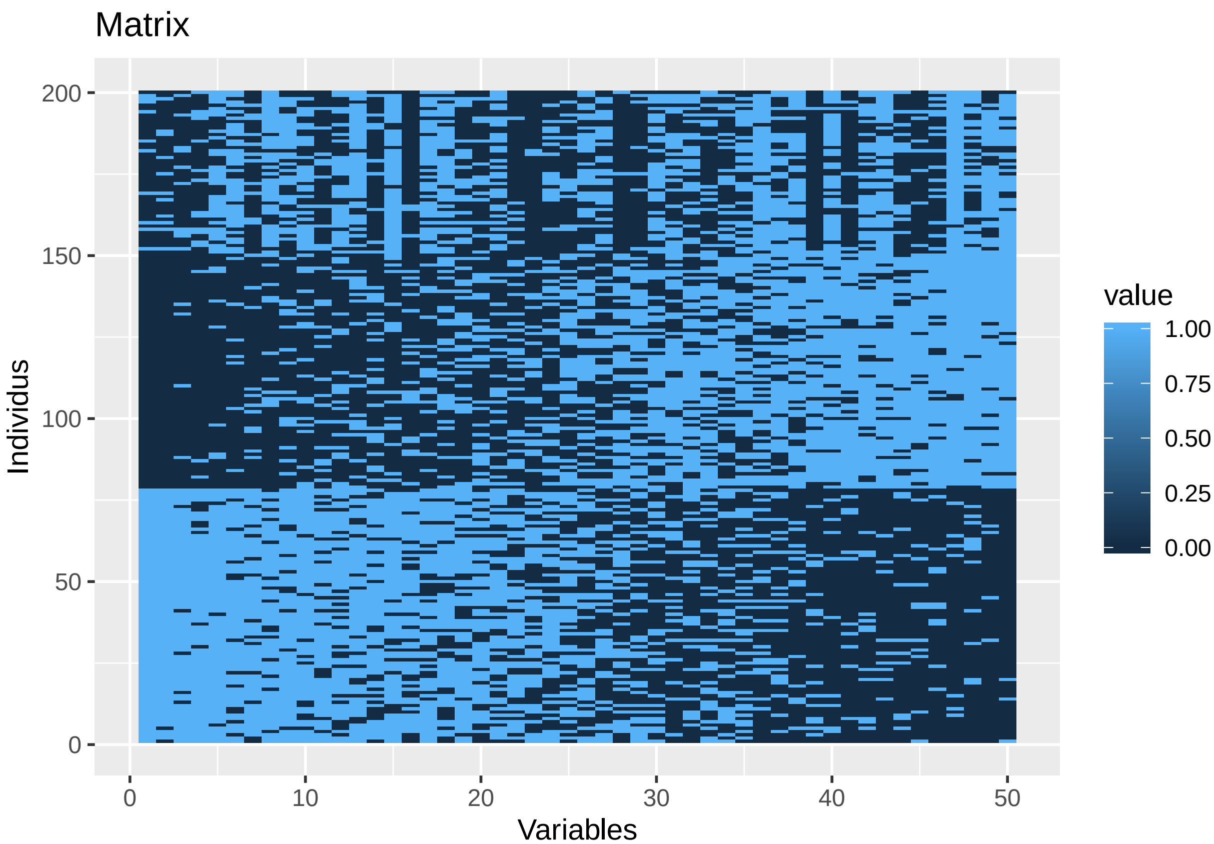
```
for(i in 1:(nks[3])){
  XsZ3[,i] <- rbernoulli(50,M[3,])
}
XsZ <- t(cbind(XsZ,XsZ3))
#Permutation des lignes
XsZ1 <- XsZ[sample(nrow(XsZ)),]
#On applique kmeans sur notre matrice XsZ1
kmeans(XsZ1,3,nstart = 10)->res.kmeans1
tidyData1 <- melt(XsZ1[order(res.kmeans1$cluster),order(M[1,])])
```

##comparaison des deux simulations

```
#simulation donnée
par(mfrow=c(1,2))
ggplot(tidyData, aes(x = Var2, y = Var1)) +
  geom_raster(aes(fill=value)) +
  scale_fill_brewer(aesthetics = "fill") +
labs(x="Variables", y="Individus", title="Matrix")
```




```
#notre simulation
ggplot(tidyData1, aes(x = Var2, y = Var1)) +
  geom_raster(aes(fill=value)) +
  labs(x="Variables", y="Individus", title="Matrix")
```



Remarques :

1. Il y'a bien 3 classes (dans le cas de notre simulation) ce qui est en accord avec le fait qu'on ait 3 composantes dans le mélange.
2. Les objets similaires sont proches les uns des autres. La couleur bleue foncée correspond à une petite distance et la couleur bleue claire indique une grande distance entre les observations. 3- La correspondance des deux tracés précedent nous conforte dans l'idée que nous avons bien simulé nos données # Exercise 2 : Equations de l'algorithme EM

Question 1:

On note: $\theta = \{\pi, M\}$

$$P(X, Z|\theta) = \prod_{i=1}^{n} P(x_i, z_i|\theta) = \prod_{i=1}^{n} P_\theta(x_i|z_i) \times P_\theta(z_i)$$

$$P(X, Z|\theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} P(x_i|\mu_k)^{z_{ik}} \times \pi_k^{z_{ik}}$$

$$P(X, Z|\theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}} (\prod_{j=1}^{p} \mu_{kj}^{x_{ij}} (1 - \mu_{kj})^{1-x_{ij}})^{z_{ik}}$$

$$\ln(P(X, Z|\theta)) = \ln(\prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}} (\prod_{j=1}^{p} \mu_{kj}^{x_{ij}} (1 - \mu_{kj})^{1-x_{ij}})^{z_{ik}})$$

$$\ln(P(X, Z|\theta)) = \ln(\prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}} (\prod_{j=1}^{p} \mu_{kj}^{x_{ij}} (1 - \mu_{kj})^{1-x_{ij}})^{z_{ik}})$$

$$\ln(P(X,Z|\theta)) = \ln(\prod_{i=1}^{n}\prod_{k=1}^{K}\pi_k^{z_{ik}}(\prod_{j=1}^{p}\mu_{kj}^{x_{ij}}(1-\mu_{kj})^{1-x_{ij}})^{z_{ik}})$$

$$\boxed{\ln(P(X,Z|\theta)) = \sum_{i=1}^{n}\sum_{k=1}^{K}(z_{ik})(\ln(\pi_k)+\sum_{j=1}^{p}x_{ij}\ln(\mu_{kj})+(1-x_{ij})\ln(1-\mu_{kj})))}$$

Question 2:

$$\begin{aligned}
t_{ik}^q &= \mathbb{E}(Z_{ik}/x_i,\theta^q)\\
&= 1\times\mathbb{P}(z_{ik}=1/x_i,\theta^q) + 0\times\mathbb{P}(z_{ik}=0/x_i,\theta^q)\\
&= \frac{\mathbb{P}(z_{ik}=1,x_i/\theta^q)}{\mathbb{P}(x_i/\theta^q)}\\
&= \frac{\mathbb{P}(z_{ik}=1,/\theta^q)\times\mathbb{P}(x_i/z_{ik}=1,/\theta^q)}{\mathbb{P}(x_i/\theta^q)}\\
&= \frac{\pi_k\times\mathbb{P}(x_i/z_{ik}=1,/\theta^q)}{\mathbb{P}(x_i/\theta^q)}\\
&= \frac{\mathbb{P}(z_{ik}=1,x_i/\theta^q)}{\mathbb{P}(x_i/\theta^q)}\\
&= \frac{\pi_k\prod\limits_{j=1}^{p}\mu_{kj}^{x_{ij}}(1-\mu_{kj})^(1-x_{ij})}{\sum\limits_{k=1}^{n}\pi_k\prod\limits_{j=1}^{p}\mu_{kj}^{x_{ij}}(1-\mu_{kj})^(1-x_{ij})}
\end{aligned}$$

Question 3:

$$\begin{aligned}
\mathbb{Q}(\theta^q/\theta) &= \mathbb{E}_{/X,\theta^q}(ln(p_n(X,Z/\theta)))\\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}_{/X,\theta^q}[Z_{ik}(\ln(P_\theta(X_i/Z_i=k))+\ln(\pi_k))]\\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}_{/X,\theta^q}(Z_{ik})(\ln(P_\theta(X_i/Z_i=k))+\ln(\pi_k))\\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^q[\ln(P_\theta(X_i/Z_i=k))+\ln(\pi_k)]\\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^q[\ln(\pi_k)+\sum_{j=1}^{p}\ln(\mu_{kj}^{x_{ij}}(1-\mu_{kj}^{1-x_{ij}}))]\\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^q[\ln(\pi_k)+\sum_{j=1}^{p}x_{ij}\ln(\mu_{kj})+(1-x_{ij})\ln(1-\mu_{kj})]
\end{aligned}$$

Question 4:

Soit k,

$$\theta^{q+1} = \underset{\theta}{\operatorname{argmax}}\,(\mathbb{Q}(\theta^q/\theta))$$

$$\frac{\partial\mathbb{Q}(\theta^q/\theta)}{\partial\mu_k} = 0 \qquad ====> \mu_k^{q+1} = \frac{\sum_{i=1}^{n}t_{ik}^q x_i}{\sum_{i=1}^{n}t_{ik}^q}$$

$$\frac{\partial \mathbb{Q}(\theta^q/\theta)}{\partial \pi_k} = 0 \qquad ====> \pi_k^{q+1} = \frac{1}{n}\sum_{i=1}^{n} t_{ik}^q$$

Question 5:

1. Etape E:
   - A cette étape, θ^q est l'estimation de θ
   - On calcule $\mathbb{Q}(\theta^q/\theta) = \mathbb{E}_{/X,\theta^q}(ln(p_n(X, Z/\theta)))$ (complete log-likelihood)
2. Etape M:
   - A cette étape, on retourne $\log(P_{\theta^{q+1}}(X))$(incomplete log-likelihood)
   - $\theta^{q+1} = \underset{\theta}{\operatorname{argmax}}\,(\mathbb{Q}(\theta^q/\theta))$

Remarque:

1.$\mathbb{E}_{/X,\theta^q}(ln(P(X/\theta))) = \mathbb{E}_{/X,\theta^q}(ln(P(X, Z/\theta))) - \mathbb{E}_{/X,\theta^q}(ln(P(Z/X, \theta)))$ Le terme $-\mathbb{E}_{/X,\theta^q}(ln(P(Z/X, \theta)))$ correspond à l'entropie de la variable latente.En maximisant la "complete log(likelihood)",on minimise l'entropie.

Question 6:

$$P_{\theta^{q+1}}(Z|X) = \frac{P_{\theta^{q+1}}(X, Z)}{P_{\theta^{q+1}}(X)}$$

$$\ln(P_{\theta^{q+1}}(Z|X)) = \ln(P_{\theta^{q+1}}(X, Z)) - \ln(P_{\theta^{q+1}}(X))$$

$$E[\ln(P_{\theta^{q+1}}(Z|X))] = E[\ln(P_{\theta^{q+1}}(X, Z)) - \ln(P_{\theta^{q+1}}(X))]$$

$$E[\ln(P_{\theta^{q+1}}(Z|X))] = E[\ln(P_{\theta^{q+1}}(X, Z)) - \ln(P_{\theta^{q+1}}(X))]$$

$$-E[\ln(P_{\theta^{q+1}}(Z|X))] = -E[\ln(P_{\theta^{q+1}}(X, Z))] + \ln(P_{\theta^{q+1}}(X))$$

$$-E[\ln(P_{\theta^{q+1}}(Z|X))] = -Q(\theta^{q+1}|\theta) + \ln(P_{\theta^{q+1}}(X))$$

Question 7:

On note:

$$\hat{\theta} = \underset{\theta\in\Theta_k}{\operatorname{argmax}}\,(\ln(P_\theta(X)))$$

$\hat{\theta}$ =la valeur maximale estimée de θ pour un model à K composantes.(Sortie de l'algorithme EM)

Θ_k=Espace de paramétres pour un melange de modéles à K composantes

$$\begin{aligned}\ln(P_{\hat{\theta}}(X)) &= \sum_{i=1}^{n} \ln(\sum_{k=1}^{K} \pi_k P_{\hat{\theta}}(X_i)) \\ &= \sum_{i=1}^{n} \ln(\sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} \mu_{kj}^{x_{ij}}(1-\mu_{kj})^{1-x_{ij}})\end{aligned}$$

Question 8:

$$\hat{K}_{BIC} = \underset{k}{\operatorname{argmax}}\,(\ln(P_{\hat{\theta}}(X)) - \frac{d_k}{2}\log(n))$$

$d_k = (k-1) + kp = k(p+1) - 1$ p est la dimension des vecteurs de l'echantillon d_k représente le nombre de paramétres à estimer et n la taille de l'echantillon (ici 200). donc:

$$\begin{aligned}\hat{K}_{BIC} &= \underset{k}{\operatorname{argmax}} \, (\ln(P_{\hat{\theta}}(X)) - \frac{k(p+1)-1}{2} \log(n)) \\ &= \underset{k}{\operatorname{argmax}} \, (\sum_{i=1}^{n} \ln(\sum_{k=1}^{K} \prod_{j=1}^{p} \pi_k \mu_{kj}^{x_{ij}} (1 - \mu_{kj}^{1-x_{ij}})) - \frac{k(p+1)-1}{2} \log(n))\end{aligned}$$

Question 9:

$$\hat{k}_{ICL} = \underset{k}{\operatorname{argmax}} \, (\ln(P_{\hat{\theta}}(X)) - d_k \ln(n) + E_{Z|X,\theta}[\ln(P_{\hat{\theta}}(Z|X))])$$

le terme$E_{Z|X,\theta}[\ln(P_{\hat{\theta}})]$ représente l'entropie de la partition

$$\hat{k}_{ICL} = \underset{k}{\operatorname{argmax}} \, (E_{Z|X,\theta}[\ln(P_{\hat{\theta}}(X,Z))] - \frac{d_k \ln(n)}{2})$$

##Question 10:

Algorithm 1 EM Algorithm

Initialize : the parameter $\theta^0 : \{\mu, \pi\}$ randomly
q<-0
while $||\theta^q - \theta^{q+1}|| > \epsilon$ **do**
  E-Step
  $\mathbb{Q}(\theta^q/\theta)$
  M-Step
  $\theta^{q+1} = \underset{\theta}{\operatorname{argmax}} \, (\mathbb{Q}(\theta^q/\theta))$
  $q \leftarrow q + 1$
end while

Exercice 3: Programmation de l'algorithme EM

Definition du vecteur Zik:

```
Zik <- matrix(0,nrow=200,ncol=K)
for( i in 1:200){
  for(k in 1:K){
    if(Z[i]==k)
    Zik[i,k]=1
  }
}
```

Question 1:

Initialisation des paramètres:

```
init.EM <- function (x,r){
  proportions <- rep(1/r,r)#initialisation des proportions
  eps <- 0.1
  n <- nrow(x)
  p <- ncol(x)
```

```
  #initilisation des vecteurs moyennes avec des valeurs random entre 0 et 1
  mu1<-matrix(runif(r*p),r,p)
  parameters=list(proportions=proportions,mu=mu1)
  return(parameters)
}
```

Etape E de l'algorithme EM : Calcul de t_{ik}

```
E.EM <- function(x,param){
  set.seed(3)
  K1<-nrow(param$mu)
  mu <- param$mu
  pi <- param$proportions
  n <- nrow(x)
  p <- ncol(x)
  t <- matrix(0,nrow=nrow(x),ncol=K1)
      for( i in 1:n){
        for(k1 in 1:K1){
          a=1
            for (j in 1:p){
              a =a*((mu[k1,j]**x[i,j])*(1-mu[k1,j])**(1-x[i,j]))

            }
            t[i,k1] <- pi[k1]*a
          }

      }
      t <- t/rowSums(t)
  return(t)

}
```

Question 2: Etape M de L'algorithme EM

```
M.EM <- function(x,param,t){
  set.seed(3)
  K1<-ncol(t)
  n <- nrow(x)
  p <- ncol(x)
  a=0
  pi <- rep(0,K1)
  for(k1 in 1:K1){
    pi[k1] <- 0
    for(i in 1:n){
        pi[k1]=pi[k1]+t[i,k1]
    }
    pi[k1] <- pi[k1]/n
  }
  mu1 <- matrix(0,K1,p)
  for(k1 in 1:K1){
    a=0
    for(i in 1:n){
      mu1[k1,] <- mu1[k1,] +t[i,k1]*x[i,]
```

```r
      a=a+t[i,k1]
    }
    mu1[k1,] <- mu1[k1,]/a
  }
parameters <-list(proportions=pi,mu=mu1)
return(parameters)

}
```

Question 3: Algortihme EM complet

```r
EM <-  function(x,k){
  set.seed(3)
  param=init.EM(x,k)
  t=E.EM(x,param)
  param.new <- M.EM(x,param,t)
  iter=0
  while((sum(abs(unlist(param.new)-unlist(param)))>(0.5e-28))){
    t=E.EM(x,param.new)
    param <- param.new
    param.new <- M.EM(x,param,t)
    iter<-iter+1

  }
  iter
  #return(c(param.new,iter))
  return(param.new)
}

EM(X,3)
```

```
## $proportions
## [1] 0.3597818 0.2502082 0.3900100
##
## $mu
##           [,1]      [,2]      [,3]      [,4]      [,5]       [,6]
## [1,] 0.1522834 0.2773602 0.1528603 0.6803677 0.3613232 0.86162190
## [2,] 0.8001384 0.6003034 0.2198318 0.5203909 0.5395607 0.09993331
## [3,] 0.8076972 0.6538551 0.8461320 0.2436092 0.4230660 0.12820184
##           [,7]         [,8]      [,9]     [,10]     [,11]     [,12]
## [1,] 0.8755115 6.360747e-91 0.2217520 0.9444207 0.3752145 0.5419788
## [2,] 0.2997780 2.197769e-01 0.9000810 0.5603515 0.3197120 0.2397835
## [3,] 0.1282018 9.743597e-01 0.8461578 0.1538681 0.5769341 0.5769342
##          [,13]     [,14]     [,15]      [,16]     [,17]     [,18]
## [1,] 0.8749186 0.2084576 0.2217580 0.26345798 0.2217670 0.2501438
## [2,] 0.1407638 0.1998369 0.6003221 0.06080017 0.7601756 0.7993056
## [3,] 0.1410220 0.8076714 0.8846183 0.70510992 0.7307763 0.8076972
##          [,19]     [,20]      [,21]      [,22]     [,23]      [,24]
## [1,] 0.8193433 0.3051408 0.80545512 0.09728168 0.6942658 0.09727803
## [2,] 0.8201780 0.1807134 0.04081369 0.73938388 0.5803396 0.63943201
## [3,] 0.1153814 0.7435705 0.20512295 0.84613190 0.2820699 0.96153945
##           [,25]     [,26]     [,27]     [,28]      [,29]     [,30]
## [1,] 0.94441177 0.6253649 0.6942720 0.6803678 0.94441115 0.5141880
```

```
## [2,] 0.98001477 0.1199205 0.7002311 0.2806306 0.78018248 0.5595525
## [3,] 0.08976717 0.3974257 0.1795083 0.3589651 0.03848618 0.4871667
##              [,31]     [,32]     [,33]     [,34]     [,35]      [,36]
## [1,] 4.732794e-72 0.8332388 0.3891166 0.6108723 0.2773603 0.31962889
## [2,] 1.997936e-01 0.7601899 0.2597956 0.5404273 0.6003437 0.03998018
## [3,] 1.000000e+00 0.1154075 0.6922897 0.3846053 0.7307502 0.57690824
##            [,37]     [,38]      [,39]     [,40]     [,41]     [,42]
## [1,] 0.04169211 0.0555888 0.90272464 0.5836787 0.6247745 0.2501463
## [2,] 0.25974344 0.1997943 0.70028120 0.3596723 0.8800969 0.5794851
## [3,] 0.89743853 0.9358991 0.05128074 0.3718112 0.2820697 0.8717982
##           [,43]     [,44]     [,45]      [,46]     [,47]     [,48]
## [1,] 0.6942627 0.3474308 0.3335339 0.62537269 0.5552934 0.3752189
## [2,] 0.4804678 0.3796464 0.1198621 0.03997591 0.6402886 0.7193732
## [3,] 0.3333246 0.7564167 0.6282149 0.44870645 0.3461708 0.7051358
##           [,49]     [,50]
## [1,] 0.8193299 0.8332482
## [2,] 0.2806464 0.6203332
## [3,] 0.1410220 0.1538422
```

#Remarques: 1. Le vecteur "proportions" et nks/200 sont proches 2. Les lignes de la matrice "mu" sont permutées mais les valeurs obtenues sont proches des valeurs initiales(matrice M initiale)

Question 4:

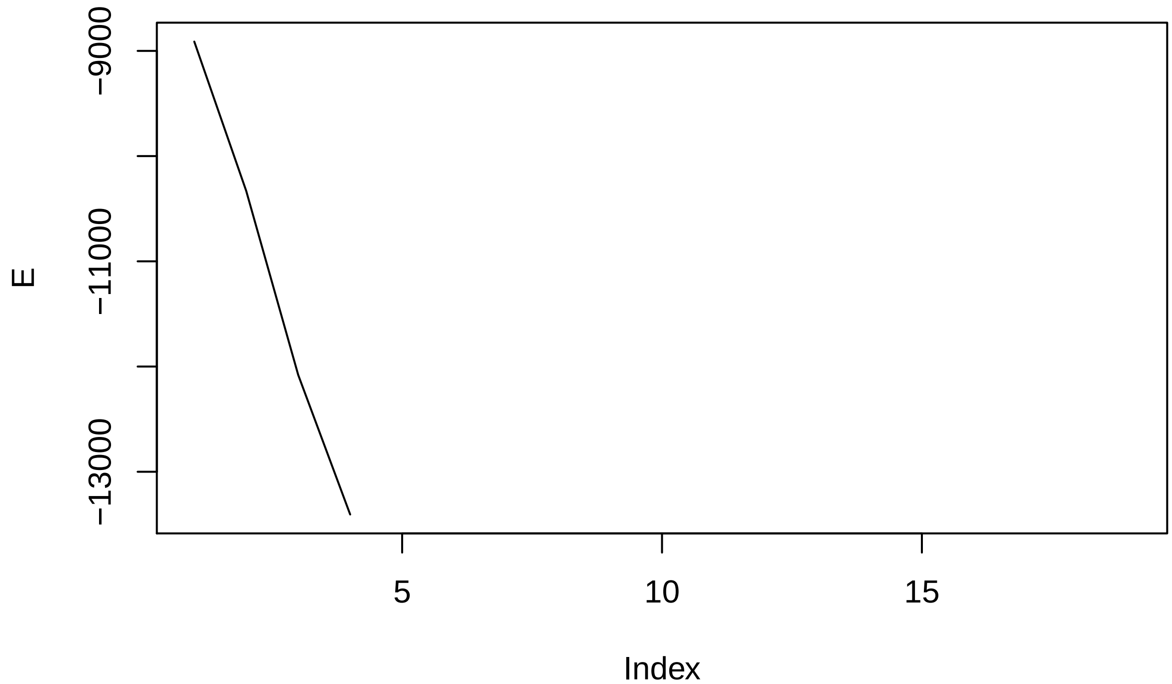
Fonction calculant la logvraisemblance completé en fonction des parametres (pour la formule voir exercice 2 question 1)

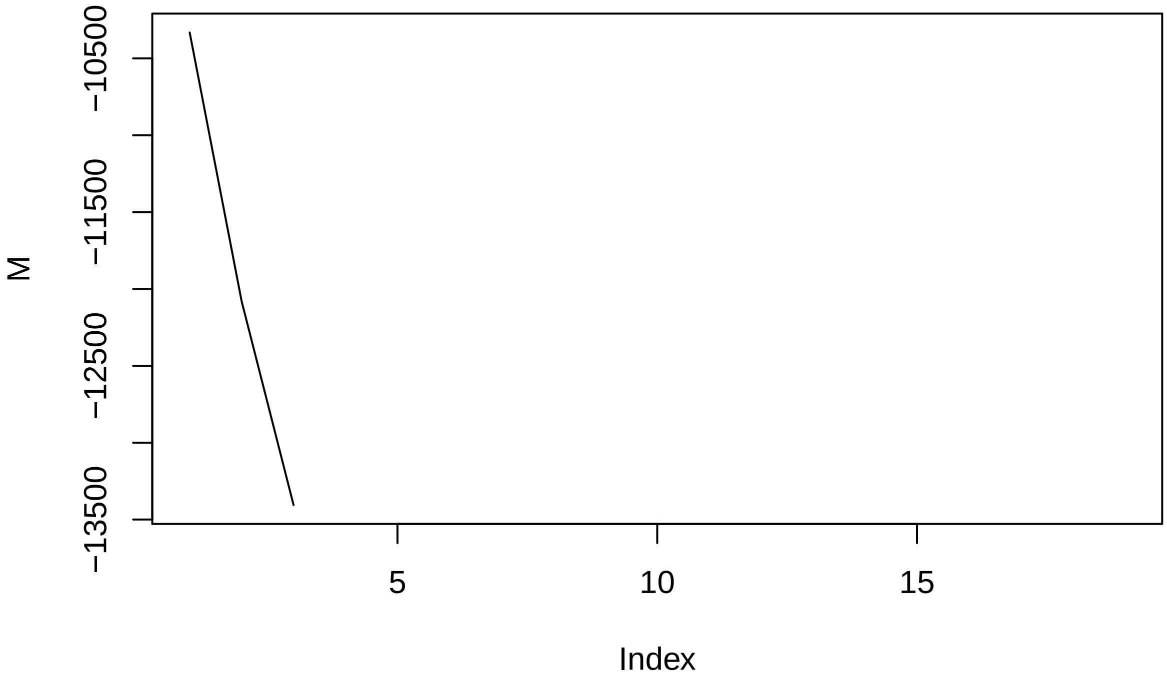
```r
log_vraisemblance <- function(parameters){
  mu <- parameters$mu
  pi <- parameters$proportions
  a=0
  for(i in 1:nrow(X)){
    b=0
    for(k in 1:nrow(parameters$mu)){
      c=0
      for (j in 1:ncol(X)){
        c=c+((X[i,j]*log(mu[k,j]))+((1-X[i,j])*log(1-mu[k,j])))
      }
      b=b+(Zik[i,k]*(log(pi[k]))+Zik[i,k]*c)
    }
    a=a+c

  }
  return(a)

}
```

#fonction tracant la logvraisemblance compléte a chaque demi-etape

```r
vraisemblance_plot <-  function(x,k){
  E=c()
  M=c()
  parameters=init.EM(x,k)
  E=c(log_vraisemblance(parameters))
  tik=E.EM(x,parameters)
  parameters.new=M.EM(x,parameters,tik)
```

```
  M <- c(log_vraisemblance(parameters.new))
  iter=0
  while((sum(abs(unlist(parameters.new)-unlist(parameters)))>(1e-28))){
    tik=E.EM(x,parameters.new)
    parameters <- parameters.new
    E <- c(E,log_vraisemblance(parameters.new))
    parameters.new <- M.EM(x,parameters,tik)
    M <- c(M,log_vraisemblance(parameters.new))
    iter<-iter+1

  }
  iter
  plot(E,type='l')
  plot(M,type='l')
}
vraisemblance_plot(X,3)
```





Le tracé n'aboutit pas du a une production de NaN que nous ne sommes pas parvenus a identifier

##Question 5:

$$\hat{K}_{BIC} = \underset{k}{\operatorname{argmax}} \left(\ln(P_{\hat{\theta}}(X)) - \frac{d_k}{2} \log(n) \right)$$

On définit la fonction

$$\ln(P_{\hat{\theta}}(X))$$

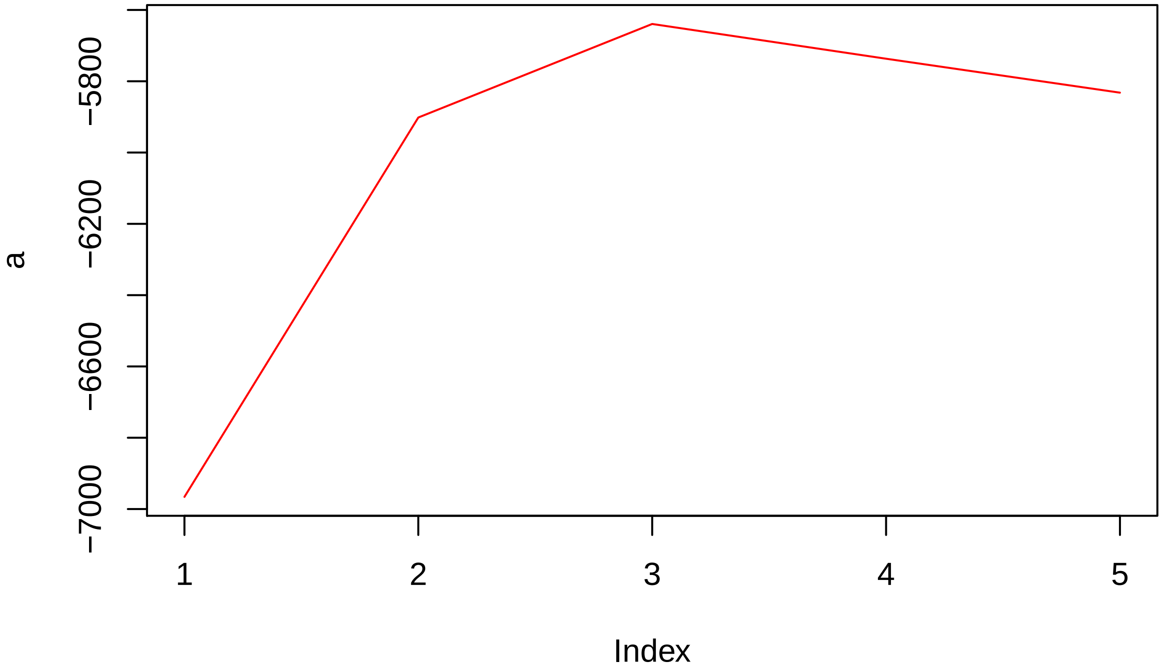
qui prend k comme paramètres et renvoie la logvraisemblance de l'echantillon X

```
log_chapeau=function(k){
  parameters <- EM(X,k)
  mu <- parameters$mu
  pi <- parameters$proportions
  a=0
  for(i in 1:nrow(X)){
    b=0
    for(l in 1:nrow(parameters$mu)){
      c=1
      for (j in 1:ncol(X)){
        c=c*(((mu[l,j])**X[i,j])*((1-mu[l,j])**(1-X[i,j])))}
      b=b+pi[l]*c}
    a=a+log(b)}
  return(a)
}
```

```
KBIC=function(k){

  bic=log_chapeau(k)-(log(n)*((k*p+k-1)/2))
  return (bic)
}
#KBIC((10))
```

#Plot de l'evolution du critére BIC en fonction de k On voit que le critére atteint son max en k egale 3 sur nos données simulés ce qui correspond bien au nombre de classe initiale de notre mixture

```
a=c()
for (i in 1:5){
  a=c(a,KBIC(i))
}
plot(a,type="l",col="red")
```




```
optimize(KBIC,lower=2,upper=4,maximum = TRUE)
```

```
## $maximum
## [1] 3
##
## $objective
## [1] -5639.367
```

K=3 correspond a une logvraisemblance maximal egale à 5639.367

##Question 6:

$$\hat{K}_{ICL} = \underset{k}{\operatorname{argmax}} \left(\mathbb{E}_{/X}[ln(P_{\hat{\theta}}(X,Z))] - \log(n)\frac{d_k}{2}\right)$$

```
log_icl_error <- function(k){
   Zik <- matrix(0,nrow=200,ncol=k)
for( i in 1:200){
  for(k1 in 1:k){
    if(Z[i]==k1)
    Zik[i,k1]=1
  }
}
  parameters <- EM(X,k)
  mu <- parameters$mu
  pi <- parameters$proportions
```

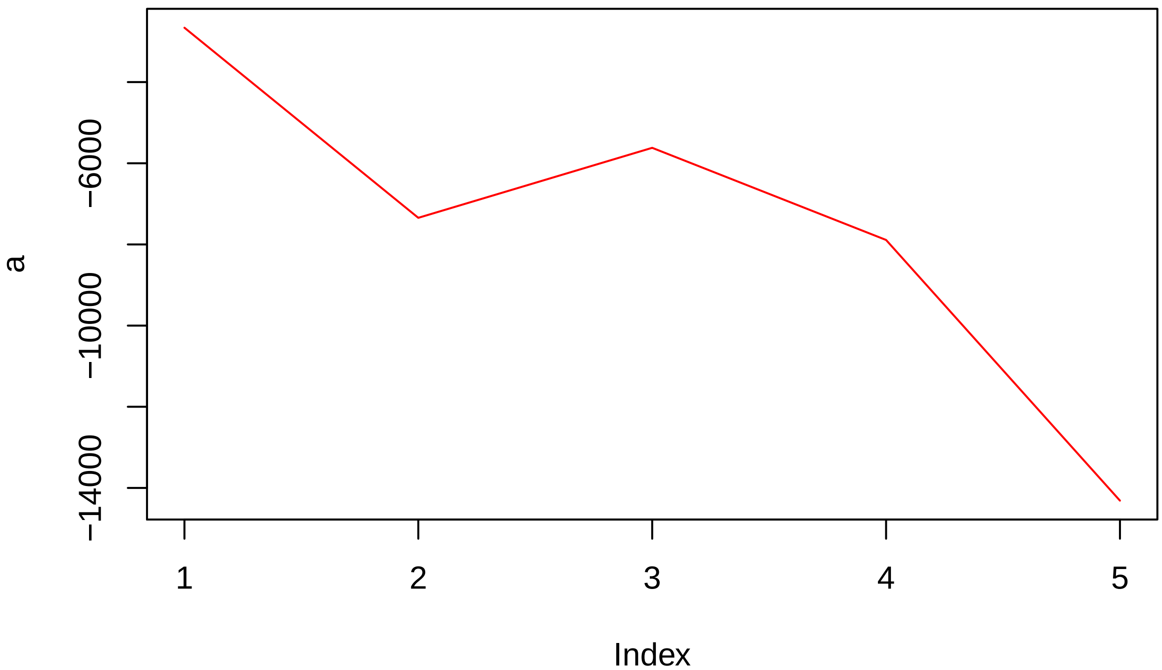
```
  a=0
  for(i in 1:nrow(X)){
    b=0
    for(k in 1:nrow(parameters$mu)){
      c=0
      for (j in 1:ncol(X)){
        c=c+(X[i,j]*log(mu[k,j]+0.001)+((1-X[i,j])*log(1-mu[k,j]-0.001))) ## Parce que tu l'as fait et

      }
      b=b+(Zik[i,k]*log(pi[k]+Zik[i,k]*c))
    }
    a=a+b
  }
  return(a)}
```

```
log_icl <- function(k){
  Zik <- matrix(0,nrow=200,ncol=k)
for( i in 1:200){
  for(k1 in 1:k){
    if(Z[i]==k1)
    Zik[i,k1]=1
  }
}
  z <- Zik
  n <- nrow(X)
  p <- ncol(X)
  param <- EM(X,k)
  mu <- param$mu
  pi <- param$proportions
  b=0
  for(i in 1:n){
    c=0
    for(l in 1:k){
      a=0
      for (j in 1:p){
        a=a+((X[i,j]*log(mu[l,j]+0.001)))+((1-X[i,j])*log(1-(mu[l,j]-0.001)))#j'ajoute 0.01 car il semb
      #print(mu[k1,j])
      }
      #print(a)
      c=c+(z[i,l]*(log(pi[l])+a))
    }

    b=b+c

  }
  return(b)
}
```

```
KICL=function(k){
  icl=mean(log_icl(k))-(log(n)*((k*p+k-1)/2))
  return (icl)
}
```

```
a=c()
```

```r
for (i in 1:5){ #ça rame si je prends 50
  a=c(a,KICL(i))
}
#a
plot(a,type="l",col="red")
```




```r
optimize(KICL,lower=2,upper=4,maximum = TRUE)
```

```
## $maximum
## [1] 3
##
## $objective
## [1] -5619.677
```

Exercice 4: Données state-firearms

Nous allons appliquer notre algorithme EM apres avoir trouvé le nombre de classes de nos données a l'aide du critére BIC implémentée précédemment. Télécharger la base de données sur https://www.kaggle.com/jboysen/state-firearms#raw_data.csv

```r
#upload de la data set
#make sure to download the dataset and replace the file path in the read.table function
statefirearms<-read.table(file="/Users/princessemame/ANAD 2019/raw_data.csv",sep=",",header=TRUE)
```

```r
#transfert des données dans une matrice de la meme forme que X( simulé plus haut )
x <- statefirearms[,3:136]
Y <- matrix(3455,1350,133)
for(i in 1:1350){
  for (j in 1:133){
    Y[i,j] =x[i,j]
  }
}
```

##Etude sur les lignes: Nous allons utlisé le critere BIC pour determiner le nombre de composantes k de notre dataset. Aprés avoir determiner k nous allons appliquer l'algorithme EM pour trouver les paramétres de notre mélange. #fonc

```
logvraisXk <- function(k){
  x <- Y
  n <- nrow(x)
  p <- ncol(x)
  param <- EM(x,k)
  K <- nrow(param$mu)
  mu <- param$mu
  pi <- param$proportions
  b=0
  for(i in 1:n){
    c=0
    for(d in 1:K){
      a=1
      for (j in 1:p){
        a=a*(((mu[d,j])**x[i,j])*((1-mu[d,j])**(1-x[i,j])))

      }
      c=c+pi[d]*a
    }
    b=b+log(c)

  }
  return(b)
}
n <- nrow(Y)
p <- ncol(Y)
critbic <- function(k){

  a <- logvraisXk(k)-(log(n)*((k*(p+1)-1)/2))

  return(-a)#on prend -a parce que la fonction optim minimise donc s on minine -a  ca veut dire quon ma
}
```

#determination du k nous avons tracé le critere bic en fonction de k sur l'intervalle 1:11. Du fait de l'imbrication de boucles et de la taille des données la compilation prend plusieurs minutes nous avons donc décidé , pour eviter d'avoir a recompiler a chaque fois de stocker le resultat obtenu dans un fichier qui vous a ete transmis en piece jointe. Si vous souhaitez proceder à la compilation il vous suffira de "decommenter" les lignes suivantes.

Il semblerait que le maximum pour le critére BIC sois obtenue avec $k = 10$ Appliquons notre algorithme EM a nos données avec k=10

```
#A <- EM(Y,10)
```

#Rseultat la compilation prend plusieurs minutes nous avons donc décidé , pour eviter d'avoir a recompiler a chaque fois de stocker les resultats obtenu dans un fichier qui vous a ete transmis en piece jointe. Si vous souhaitez proceder à la compilation il vous suffira de "decommenter" les lignes suivantes.

```
#write.table(A$mu, file = "Vecteur moyenne", append = FALSE, quote = TRUE, sep = ";")
#write.table(A$proportions, file = "Vecteur proportions", append = FALSE, quote = TRUE, sep = ";")
```

#a matrice des vecteurs moyennes

```r
vecteur_moyene <- read.table(file = "/Users/princessemame/ANAD 2019/Vecteur moyenne",sep=";",header=TRU
vecteur_moyene
```

```
##           V1         V2        V3         V4        V5        V6        V7
## 1  0.2319806 0.07335781 0.0000000 0.03286215 0.0000000 0.0000000 0.0000000
## 2  0.0000000 0.37777778 0.3777778 0.91111111 0.0000000 0.0000000 0.0000000
## 3  0.3157895 0.39473684 0.3157895 0.31578947 0.0000000 0.3157895 0.3157895
## 4  0.4067797 1.00000000 0.5423729 1.00000000 0.0000000 0.0000000 0.0000000
## 5  0.2695646 0.66087033 0.1999995 0.19999955 0.0000000 0.0000000 0.0000000
## 6  0.0000000 0.00000000 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000
## 7  0.0000000 0.00000000 1.0000000 1.00000000 0.0000000 0.0000000 0.0000000
## 8  0.1292939 0.28211792 0.0565248 0.16058185 0.0000000 0.0000000 0.0000000
## 9  0.7804878 0.87804878 0.7804878 0.78048780 0.4878049 0.2926829 0.2926829
## 10 0.5510204 0.55102041 0.3469388 0.34693878 0.0000000 0.0000000 0.0000000
##            V8            V9        V10        V11        V12        V13
## 1  0.24894693 1.183305e-139 0.00000000 0.00000000 0.00000000 0.10954077
## 2  0.08888889  0.000000e+00 0.37777778 0.37777778 0.00000000 0.00000000
## 3  0.00000000  1.000000e+00 0.00000000 0.00000000 0.00000000 0.00000000
## 4  0.84745763  3.898305e-01 0.94915254 0.89830508 0.08474576 0.00000000
## 5  0.00000000  2.608690e-02 0.06956506 0.19999955 0.02608690 0.06086943
## 6  0.00000000  0.000000e+00 0.00000000 0.00000000 0.00000000 0.00000000
## 7  0.00000000  0.000000e+00 0.00000000 0.00000000 0.00000000 0.00000000
## 8  0.11632184  3.982429e-02 0.04496291 0.04496291 0.00000000 0.00000000
## 9  0.00000000  8.780488e-01 0.58536585 0.48780488 0.70731707 0.58536585
## 10 0.00000000  0.000000e+00 0.00000000 0.20408163 0.44897959 0.00000000
##           V14       V15          V16       V17       V18        V19
## 1  0.00000000 0.0000000 3.350225e-56 0.0000000 0.0000000 0.00000000
## 2  0.00000000 0.0000000 4.444444e-01 0.0000000 0.0000000 0.00000000
## 3  0.00000000 0.0000000 3.947368e-01 0.0000000 0.0000000 0.00000000
## 4  0.08474576 0.1186441 4.915254e-01 0.5423729 0.5423729 0.54237288
## 5  0.02608690 0.0000000 3.043471e-01 0.2086952 0.2086952 0.04347816
## 6  0.00000000 0.0000000 0.000000e+00 0.0000000 0.0000000 0.00000000
## 7  0.00000000 0.0000000 0.000000e+00 1.0000000 1.0000000 1.00000000
## 8  0.00000000 0.0000000 3.725498e-02 0.0000000 0.0000000 0.00000000
## 9  0.70731707 0.0000000 1.000000e+00 0.4634146 0.4878049 0.00000000
## 10 0.44897959 0.0000000 6.530612e-01 0.5510204 0.5510204 0.55102041
##           V20        V21        V22        V23         V24        V25
## 1  0.00000000 0.00000000 0.00000000 0.00000000 0.000000000 0.00000000
## 2  0.00000000 0.48888889 1.00000000 1.00000000 0.533333333 0.46666667
## 3  0.00000000 0.00000000 0.97368421 0.94736842 0.578947368 0.97368421
## 4  0.54237288 0.54237288 0.93220339 0.93220339 0.457627119 0.08474576
## 5  0.09565196 0.69565059 0.06956506 0.06956506 0.000000000 0.04347816
## 6  0.00000000 1.00000000 1.00000000 1.00000000 1.000000000 0.00000000
## 7  1.00000000 1.00000000 1.00000000 1.00000000 0.000000000 0.00000000
## 8  0.00000000 0.06037877 0.16572045 0.08992583 0.002569309 0.03211637
## 9  0.00000000 1.00000000 0.90243902 0.78048780 0.487804878 0.78048780
## 10 0.00000000 1.00000000 0.79591837 0.53061224 0.000000000 0.53061224
##            V26        V27        V28       V29        V30       V31
## 1  0.000000000 0.00000000 0.00000000 0.5602070 0.03558343 0.3353339
## 2  0.466666667 0.46666667 1.00000000 0.8222222 0.00000000 0.4666667
## 3  0.605263158 0.36842105 0.97368421 0.6052632 0.39473684 0.6052632
## 4  0.084745763 0.00000000 0.93220339 0.5423729 0.00000000 0.5423729
## 5  0.043478162 0.04347816 0.06956506 1.0000000 0.30434714 1.0000000
## 6  0.000000000 0.00000000 1.00000000 1.0000000 0.52173913 1.0000000
```

```
## 7  0.000000000 0.00000000 1.00000000 1.0000000 0.00000000 1.0000000
## 8  0.002569309 0.00000000 0.16572045 0.8886173 0.19591390 0.7165474
## 9  0.487804878 0.78048780 0.90243902 1.0000000 0.41463415 1.0000000
## 10 0.000000000 0.00000000 0.79591837 1.0000000 0.71428571 1.0000000
##          V32       V33        V34       V35         V36        V37
## 1  0.6647300 0.1605637 0.10015711 0.5371147 0.109540772 0.30953343
## 2  0.0000000 0.0000000 0.00000000 1.0000000 0.000000000 1.00000000
## 3  0.6052632 0.0000000 0.00000000 0.6052632 0.394736842 0.39473684
## 4  0.4067797 0.4576271 0.05084746 1.0000000 0.915254237 1.00000000
## 5  0.8782589 0.4608685 0.26086897 0.6956506 0.260868973 0.56521837
## 6  0.2173913 1.0000000 1.00000000 0.0000000 0.086956522 0.08695652
## 7  0.0000000 0.0000000 0.00000000 0.4736842 0.000000000 1.00000000
## 8  0.7112276 0.3618669 0.30281015 0.2028898 0.005138619 0.15863557
## 9  0.6097561 0.7073171 0.70731707 1.0000000 0.878048780 0.87804878
## 10 1.0000000 1.0000000 1.00000000 1.0000000 0.551020408 0.55102041
##           V38        V39        V40       V41        V42        V43
## 1  0.00000000 0.02190673 0.02738519 0.3366888 0.05477062 0.00000000
## 2  0.00000000 0.22222222 0.91111111 0.8000000 0.24444444 0.00000000
## 3  0.31578947 0.39473684 1.00000000 0.4473684 0.26315789 0.10526316
## 4  0.06779661 0.45762712 0.08474576 0.3898305 0.38983051 0.08474576
## 5  0.02608690 0.77391354 0.50434668 0.3304340 0.11304322 0.00000000
## 6  0.00000000 0.00000000 0.00000000 0.6521739 0.65217391 0.00000000
## 7  0.00000000 0.00000000 1.00000000 1.0000000 0.73684211 0.00000000
## 8  0.00000000 0.00000000 0.03468568 0.1342814 0.08093319 0.00000000
## 9  0.87804878 0.87804878 1.00000000 1.0000000 1.00000000 0.58536585
## 10 0.00000000 0.55102041 0.32653061 0.6530612 0.65306122 0.02040816
##           V44        V45        V46        V47        V48        V49
## 1  0.15883412 0.00000000 0.04929335 0.48789630 0.15883405 0.04929328
## 2  0.08888889 0.00000000 0.08888889 0.97777778 0.00000000 0.00000000
## 3  0.44736842 0.26315789 0.44736842 0.60526316 0.31578947 0.15789474
## 4  0.38983051 0.38983051 0.38983051 0.08474576 0.33898305 0.33898305
## 5  0.31304277 0.11304322 0.27826024 0.74782665 0.22608644 0.08695632
## 6  0.65217391 0.65217391 0.65217391 1.00000000 0.65217391 0.65217391
## 7  1.00000000 0.73684211 1.00000000 1.00000000 0.00000000 0.00000000
## 8  0.06166342 0.04367826 0.05138619 0.81950616 0.02312380 0.02312380
## 9  1.00000000 1.00000000 1.00000000 0.70731707 1.00000000 1.00000000
## 10 0.38775510 0.38775510 0.38775510 1.00000000 0.06122449 0.06122449
##           V50        V51        V52       V53         V54        V55
## 1  0.10954077 0.00000000 0.00000000 0.5973249 0.158833214 0.00000000
## 2  0.00000000 0.00000000 0.00000000 0.9111111 0.000000000 0.11111111
## 3  0.31578947 0.15789474 0.31578947 1.0000000 0.394736842 0.00000000
## 4  0.33898305 0.33898305 0.33898305 0.5423729 0.186440678 1.00000000
## 5  0.22608644 0.08695632 0.19130391 0.7652157 0.486955420 0.39130347
## 6  0.65217391 0.65217391 0.65217391 1.0000000 0.000000000 0.00000000
## 7  0.00000000 0.00000000 0.00000000 1.0000000 1.000000000 0.73684211
## 8  0.00000000 0.00000000 0.00000000 0.6486737 0.008992794 0.01670051
## 9  1.00000000 1.00000000 1.00000000 1.0000000 0.878048780 0.12195122
## 10 0.06122449 0.06122449 0.06122449 0.4489796 0.551020408 0.16326531
##           V56        V57           V58       V59       V60       V61
## 1  0.00000000 0.00000000 1.443987e-230 0.4066002 0.0000000 0.1984987
## 2  0.48888889 0.00000000  0.000000e+00 0.4666667 0.0000000 0.0000000
## 3  0.00000000 0.00000000  0.000000e+00 1.0000000 0.0000000 0.0000000
## 4  1.00000000 0.03389831  1.186441e-01 0.8135593 0.0000000 0.3050847
## 5  0.61738991 0.01739126  1.739126e-02 0.6000009 0.0000000 0.2000018
```

```
## 6  0.00000000 0.00000000  0.000000e+00 0.3043478 0.0000000 0.0000000
## 7  1.00000000 0.00000000  7.368421e-01 1.0000000 0.0000000 0.0000000
## 8  0.01670051 0.00000000  1.541586e-02 0.4094997 0.0000000 0.1538474
## 9  0.12195122 0.00000000  0.000000e+00 1.0000000 0.0000000 0.2682927
## 10 0.16326531 0.00000000  0.000000e+00 1.0000000 0.2857143 0.6938776
##          V62       V63        V64         V65          V66       V67
## 1  0.0000000 0.5316377 0.39646817 0.005265423 8.439120e-63 0.0000000
## 2  0.0000000 1.0000000 0.08888889 0.466666667 0.000000e+00 0.2888889
## 3  0.0000000 1.0000000 0.00000000 1.000000000 0.000000e+00 0.0000000
## 4  0.4576271 1.0000000 0.54237288 0.288135593 8.474576e-02 0.6610169
## 5  0.0000000 0.6956506 0.23478208 0.043478162 1.999995e-01 0.5391292
## 6  0.0000000 0.0000000 0.00000000 0.000000000 0.000000e+00 0.0000000
## 7  0.0000000 1.0000000 0.42105263 0.000000000 7.368421e-01 1.0000000
## 8  0.0000000 0.2388602 0.07915101 0.032166001 0.000000e+00 0.0000000
## 9  0.4878049 1.0000000 0.90243902 0.902439024 0.000000e+00 0.6097561
## 10 0.0000000 1.0000000 1.00000000 0.448979592 0.000000e+00 0.5102041
##          V68       V69       V70        V71       V72        V73       V74
## 1  0.0000000 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.1436724
## 2  0.0000000 0.0000000 0.0000000 0.11111111 0.4222222 0.00000000 1.0000000
## 3  0.0000000 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.6052632
## 4  0.0000000 0.3559322 0.2711864 0.45762712 0.3898305 0.01694915 1.0000000
## 5  0.0000000 0.0000000 0.1565214 0.15652138 0.2434777 0.04347816 0.4173904
## 6  0.0000000 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.0000000
## 7  0.0000000 0.7894737 0.0000000 0.31578947 0.0000000 0.00000000 1.0000000
## 8  0.0000000 0.0000000 0.0000000 0.03468568 0.0000000 0.00000000 0.2026777
## 9  0.4878049 0.4878049 0.4878049 0.60975610 0.4878049 0.00000000 0.8780488
## 10 0.0000000 0.3673469 0.0000000 0.20408163 0.5510204 0.55102041 1.0000000
##          V75        V76        V77       V78        V79        V80
## 1  0.3567614 0.29547268 0.10954077 0.0000000 0.04929335 0.00000000
## 2  0.8000000 0.40000000 0.00000000 0.0000000 0.00000000 0.00000000
## 3  1.0000000 1.00000000 0.26315789 0.1052632 1.00000000 1.00000000
## 4  0.5423729 0.54237288 0.00000000 0.0000000 0.35593220 0.35593220
## 5  0.4782598 0.31304277 0.03478253 0.0000000 0.18260828 0.03478253
## 6  0.0000000 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000
## 7  1.0000000 1.00000000 0.00000000 0.0000000 1.00000000 1.00000000
## 8  0.1013109 0.02447585 0.00000000 0.0000000 0.01027724 0.00000000
## 9  0.9024390 0.90243902 0.29268293 0.0000000 0.90243902 0.90243902
## 10 0.5918367 0.59183673 0.02040816 0.0000000 0.02040816 0.02040816
##           V81       V82        V83       V84        V85        V86
## 1  0.04929335 0.0000000 0.00000000 0.8231126 0.00000000 0.00000000
## 2  0.00000000 0.1111111 0.28888889 1.0000000 0.00000000 0.37777778
## 3  1.00000000 0.0000000 0.00000000 1.0000000 0.00000000 0.00000000
## 4  0.35593220 0.5423729 0.08474576 1.0000000 0.38983051 0.32203390
## 5  0.18260828 0.5304336 0.11304322 0.8347830 0.04347816 0.04347816
## 6  0.00000000 0.0000000 0.00000000 0.6956522 0.00000000 0.00000000
## 7  1.00000000 0.7894737 0.00000000 1.0000000 0.00000000 0.00000000
## 8  0.00000000 0.0000000 0.00000000 0.7485882 0.00000000 0.04110895
## 9  0.78048780 0.6097561 0.00000000 1.0000000 0.00000000 0.00000000
## 10 0.02040816 0.4285714 0.00000000 1.0000000 0.00000000 0.16326531
##           V87        V88       V89        V90       V91       V92
## 1  0.00000000 0.00000000 0.1153133 0.10954077 0.0000000 0.7941858
## 2  0.00000000 0.00000000 0.4666667 0.00000000 0.0000000 1.0000000
## 3  0.00000000 0.39473684 1.0000000 1.00000000 0.3947368 1.0000000
## 4  0.01694915 0.54237288 1.0000000 0.54237288 0.5762712 1.0000000
```

```
## 5  0.23478207 0.05217379 0.3304340 0.05217379 0.0260869 1.0000000
## 6  0.00000000 0.00000000 0.0000000 0.00000000 0.0000000 1.0000000
## 7  0.00000000 1.00000000 1.0000000 1.00000000 0.0000000 1.0000000
## 8  0.10919565 0.04239360 0.2928320 0.08093324 0.0000000 0.9544945
## 9  0.00000000 1.00000000 1.0000000 1.00000000 1.0000000 1.0000000
## 10 0.44897959 0.44897959 1.0000000 1.00000000 1.0000000 1.0000000
##          V93       V94        V95        V96        V97           V98
## 1  0.0000000 0.0000000 0.00000000 0.00000000 0.04929335 2.166776e-313
## 2  0.3777778 0.3777778 0.28888889 0.00000000 0.00000000  0.000000e+00
## 3  0.3947368 0.3947368 0.00000000 0.39473684 0.39473684  3.947368e-01
## 4  0.9491525 0.5423729 0.00000000 0.08474576 0.54237288  5.423729e-01
## 5  0.5217379 0.5217379 0.06086943 0.28695587 0.28695587  2.869559e-01
## 6  1.0000000 1.0000000 0.00000000 0.04347826 0.04347826  4.347826e-02
## 7  0.9473684 0.9473684 0.00000000 1.00000000 1.00000000  1.000000e+00
## 8  0.1169036 0.1002031 0.00000000 0.11561892 0.15287390  1.528739e-01
## 9  1.0000000 1.0000000 0.00000000 0.87804878 1.00000000  8.780488e-01
## 10 1.0000000 1.0000000 0.32653061 1.00000000 1.00000000  1.000000e+00
##            V99       V100       V101          V102      V103      V104
## 1  0.333884148 0.00000000 0.07120008  2.738519e-01 0.7457923 0.0000000
## 2  0.466666667 0.11111111 0.48888889  5.333333e-01 0.9111111 0.0000000
## 3  0.394736842 0.00000000 0.00000000  0.000000e+00 0.0000000 0.3157895
## 4  0.542372881 1.00000000 1.00000000  1.000000e+00 1.0000000 0.0000000
## 5  0.773913550 0.05217379 0.65217469  6.260855e-01 1.0000000 0.0000000
## 6  0.000000000 0.00000000 0.00000000  0.000000e+00 1.0000000 0.0000000
## 7  1.000000000 1.00000000 1.00000000  1.000000e+00 1.0000000 0.0000000
## 8  0.005188761 0.00000000 0.00000000 1.155551e-198 0.1038421 0.0000000
## 9  0.878048780 0.12195122 0.12195122  7.073171e-01 0.7073171 0.2926829
## 10 0.551020408 0.44897959 1.00000000  4.489796e-01 1.0000000 0.0000000
##         V105       V106       V107      V108       V109      V110
## 1  0.0000000 0.00000000 0.07120008 0.1095408 0.44342492 0.0000000
## 2  0.0000000 0.00000000 0.37777778 0.0000000 0.46666667 0.0000000
## 3  0.3947368 0.31578947 0.39473684 0.3157895 0.39473684 0.0000000
## 4  0.0000000 0.54237288 0.54237288 0.5423729 0.54237288 0.0000000
## 5  0.4695641 0.07826069 0.59130526 0.3913035 0.95652183 0.0260869
## 6  0.0000000 0.00000000 0.00000000 0.0000000 0.00000000 0.0000000
## 7  0.0000000 1.00000000 1.00000000 1.0000000 1.00000000 0.0000000
## 8  0.0000000 0.00000000 0.00000000 0.0000000 0.01803531 0.0000000
## 9  0.2926829 0.87804878 0.87804878 0.8780488 0.87804878 0.5853659
## 10 0.0000000 0.00000000 0.55102041 0.0000000 0.55102041 0.0000000
##          V111      V112      V113       V114       V115      V116
## 1  0.25731039 0.1436724 0.2680116 0.00000000 0.00000000 0.0000000
## 2  0.00000000 1.0000000 0.8666667 0.00000000 0.37777778 0.0000000
## 3  0.60526316 0.5263158 0.5789474 0.00000000 0.00000000 0.0000000
## 4  1.00000000 0.9152542 0.4915254 1.00000000 1.00000000 0.4067797
## 5  0.23478433 0.3652166 0.4695642 0.26086898 0.72173750 0.0260869
## 6  0.00000000 0.0000000 0.0000000 0.00000000 1.00000000 0.0000000
## 7  1.00000000 0.0000000 1.0000000 1.00000000 1.00000000 0.0000000
## 8  0.01287211 0.1731306 0.1169888 0.01541585 0.08478721 0.0000000
## 9  0.58536585 0.2926829 0.7560976 1.00000000 1.00000000 0.5853659
## 10 0.55102041 1.0000000 0.9387755 1.00000000 1.00000000 0.0000000
##         V117      V118      V119       V120      V121       V122
## 1  0.0000000 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000
## 2  0.2222222 0.0000000 0.0000000 0.42222222 0.0000000 0.53333333
## 3  0.0000000 0.0000000 0.0000000 0.00000000 0.0000000 0.07894737
```

```
## 4  0.4067797 0.3898305 0.4067797 0.91525424 0.4067797 0.49152542
## 5  0.1565214 0.1999995 0.0260869 0.28695587 0.0260869 0.23478208
## 6  0.0000000 0.0000000 0.0000000 1.00000000 0.0000000 0.00000000
## 7  0.0000000 0.0000000 0.0000000 0.00000000 0.0000000 1.00000000
## 8  0.0000000 0.0000000 0.0000000 0.02569309 0.0000000 0.18627492
## 9  0.5853659 0.4878049 0.5853659 1.00000000 0.5853659 0.70731707
## 10 0.0000000 0.0000000 0.0000000 0.44897959 0.0000000 0.26530612
##          V123      V124        V125        V126        V127      V128
## 1  0.00000000 0.0000000 0.000000000 0.000000000 0.000000000 0.0000000
## 2  0.53333333 0.2444444 0.111111111 0.488888889 0.111111111 0.4888889
## 3  0.07894737 0.2894737 0.000000000 0.000000000 0.000000000 0.0000000
## 4  0.49152542 0.9491525 1.000000000 1.000000000 1.000000000 1.0000000
## 5  0.23478208 0.2260864 0.147825751 0.373912193 0.173912648 0.6347812
## 6  0.00000000 0.0000000 0.000000000 0.000000000 0.000000000 1.0000000
## 7  1.00000000 0.9473684 0.736842105 1.000000000 0.736842105 1.0000000
## 8  0.22609922 0.0000000 0.001284655 0.001284655 0.001284655 0.0963491
## 9  0.70731707 0.7804878 0.000000000 0.000000000 1.000000000 1.0000000
## 10 0.26530612 0.0000000 0.000000000 0.000000000 1.000000000 1.0000000
##          V129      V130       V131      V132       V133
## 1  0.00000000 0.0000000 0.32404405 0.0000000 0.07646550
## 2  0.04444444 0.2222222 0.75555556 0.0000000 0.46666667
## 3  0.39473684 0.3947368 1.00000000 0.3947368 1.00000000
## 4  0.54237288 0.5423729 0.54237288 0.9152542 0.91525424
## 5  0.27826024 0.2782602 0.63478117 0.0000000 0.24347997
## 6  0.00000000 0.0000000 0.00000000 0.0000000 0.00000000
## 7  0.00000000 0.0000000 0.00000000 0.0000000 0.00000000
## 8  0.00000000 0.0000000 0.07943778 0.0000000 0.04501255
## 9  0.29268293 0.2926829 0.90243902 0.4146341 0.41463415
## 10 0.00000000 0.0000000 0.89795918 0.4489796 1.00000000
```

#vecteur des propotions

```r
vecteur_proportions <- read.table(file = "/Users/princessemame/ANAD 2019/Vecteur proportions",sep=";",h
vecteur_proportions
```

```
##             x
## 1  0.13524475
## 2  0.03333333
## 3  0.02814815
## 4  0.04370370
## 5  0.08518538
## 6  0.01703704
## 7  0.01407407
## 8  0.57660690
## 9  0.03037037
## 10 0.03629630
```