Mélange de Bernoulli

24/10/2018

Modèle

Considérons un vecteur aléatoire binaire $\boldsymbol{x} \in [0,1]^p$ de p variables x_j suivant chacune une distribution de Bernoulli $\mathcal{B}(\mu_j)$. La distribution du vecteur s'exprime comme:

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = \prod_{j=1}^{p} \mu_j^{x_j}(1-\mu_j)^{1-x_j},$$

avec $\boldsymbol{x} = (x_1, \cdots, x_p)^T$ et $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_p)^T$.

Soit une distribution mélange à K composantes de Bernoulli

$$p(\boldsymbol{x}|\boldsymbol{\pi}, \boldsymbol{M}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)$$

où les π_k sont les proportions du mélange et les $p(\boldsymbol{x}|\boldsymbol{\mu}_k)$ sont des distributions de Bernoulli multivariées de paramètres $\boldsymbol{\mu}_k = (\mu_{k1}, \cdots, \mu_{kp})^T$, et $M = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}^T$ la matrice des paramètres des densités de classes.

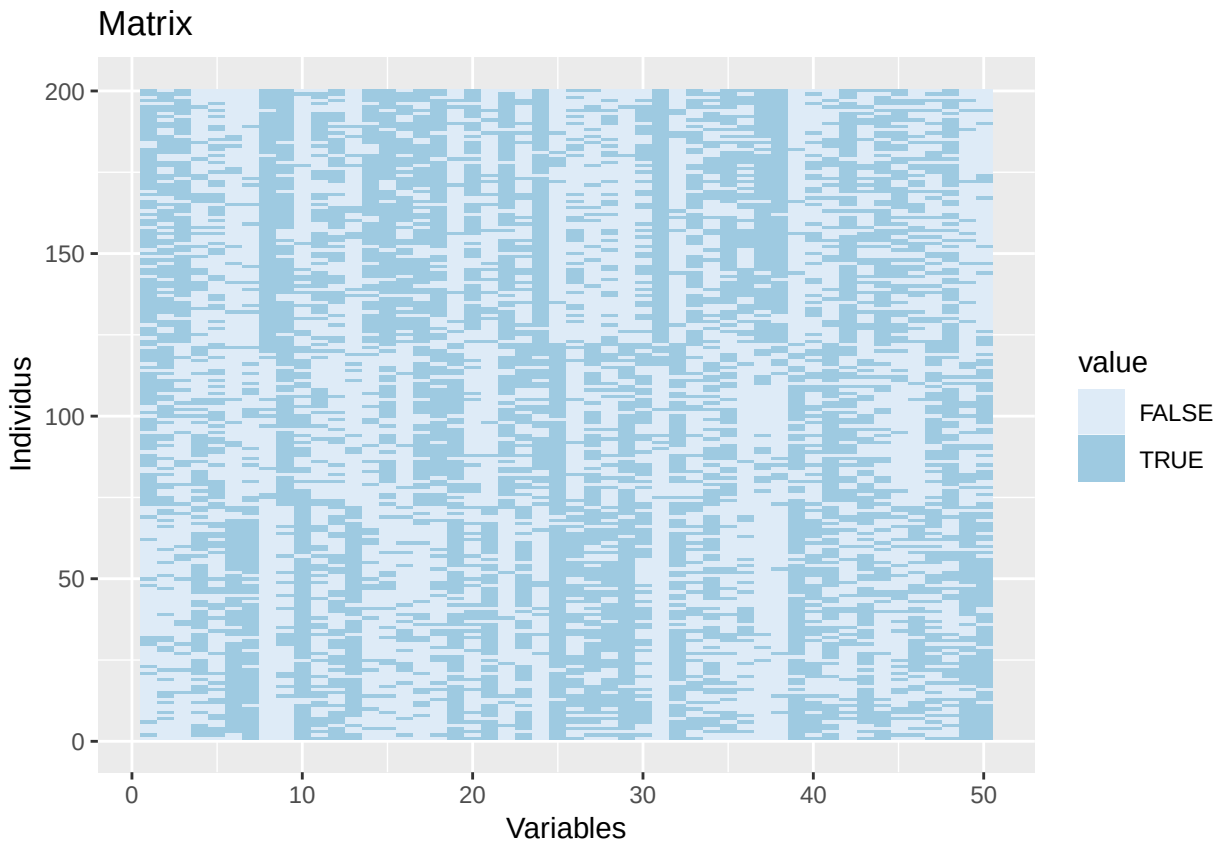
Dans la suite nous considérerons

- un échantillon observé $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ issu de cette distribution mélange,
- des variables latentes $Z = \{z_1, \cdots, z_n\}$ indiquant la composante d'origine de chaque $\boldsymbol{x}_i$.

Simulation

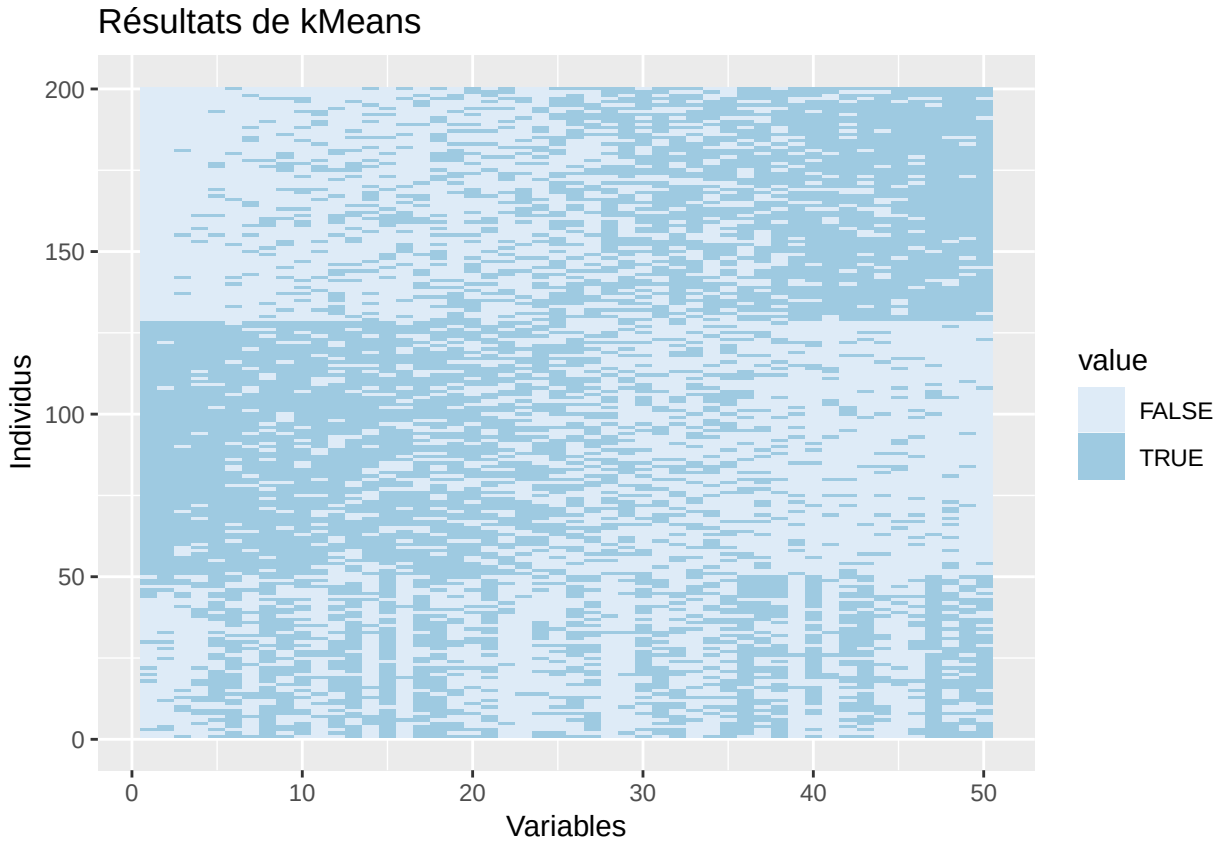
```
set.seed(3)
K<-3
p<-50
n<-200
pi<-c(1/3,1/3,1/3)
M<-matrix(runif(K*p),K,p)
M[K,]<-1-M[1,]
nks<-rmultinom(1,200,prob = pi)
Z<-rep(1:length(nks),nks)
X <-do.call(rbind,
                  mapply(function(nk,k){
                    matrix(rbernoulli(nk*p,p=M[k,]),
                           nrow = nk,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

real_X <- melt(X)
ggplot(real_X, aes(x = Var2, y = Var1)) +
  geom_raster(aes(fill=value)) +
  scale_fill_brewer(aesthetics = "fill") +
  labs(x="Variables", y="Individus", title="Matrix")
```




```
kmeans(X,3,nstart = 10)->res.kmeans
# réorganiser selon les paramètres de la première composante
tidyData<-melt(X[order(res.kmeans$cluster),order(M[1,])])

ggplot(tidyData, aes(x = Var2, y = Var1)) +
  geom_raster(aes(fill=value)) +
  scale_fill_brewer(aesthetics = "fill") +
  labs(x="Variables", y="Individus", title="Résultats de kMeans")
```



Selon les deux matrix en-dessous nous avons constaté que:

- Dans la première matrice, puisque la somme des paramètres du troisième composant et des paramètres correspondants de la première partie est 1, nous voyons que les valeurs de la première partie et de la troisième partie sont exactement opposées;
- Nous avons vérifié les résultats de K-means et la méthode a bien trouvé les bons cluters;
- Dans la deuxième matrice, les colonnes de la matrice sont disposées dans l'ordre croissant des paramètres de la première composante, et les lignes sont de l'ordre cluster2, 3 et 1.

Exo 2 Équations de l'algorithme EM

1.

D'abord nous avons:

$$p(X|Z,\theta) = \prod_{i=1}^{N}\prod_{k=1}^{K} p(X_i|Z_i = k, \mu_k)^{z_{ik}}$$

aussi

$$p(Z|\theta) = \prod_{i=1}^{N}\prod_{k=1}^{K} \pi_k^{z_{ik}}$$

nous avons donc

$$p(X,Z|\theta) = p(X|Z,\theta) \times p(Z|\theta) = \prod_{i=1}^{N}\prod_{k=1}^{K} \pi_k p(X_i|Z_i = k, \mu_k)^{z_{ik}}$$

Et enfin nou avons

$$\begin{aligned}\ln p(X, Z|\theta) &= \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\{\ln\pi_k + \ln p(X_i|Z_i = k, \mu_k)\} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\{\ln\pi_k + \ln f_k(X_i|\mu_k)\}\end{aligned}$$

2.

$$\begin{aligned}{\tau_{ik}}^q &= \mathbb{E}[Z_{ik}|X_i, \theta^q] \\ &= \mathbb{E}[1_{z_i=k}|X_i, \theta^q] \\ &= \mathbb{P}(z_i = k|X_i, \theta^q) \\ &= \frac{\mathbb{P}(z_i = k, X_i|\theta^q)}{\mathbb{P}(X_i|\theta^q)} \\ &= \frac{\pi_k f_k(X_i|\mu_k)}{f(X_i|\theta^q)} \\ &= \frac{\pi_k f_k(X_i|\mu_k)}{\sum_{l=1}^{K}\pi_l f_l(X_i|\mu_l)} \\ &= \frac{\pi_k \prod_{j=1}^{p}\mu_{kj}^{x_{ij}}(1-\mu_{kj})^{(1-x_{ij})}}{\sum_{l=1}^{K}\pi_l \prod_{j=1}^{p}\mu_{lj}^{x_{ij}}(1-\mu_{lj})^{(1-x_{ij})}}\end{aligned}$$

3.

$$\begin{aligned}Q(\theta^q|\theta^{q-1}) &= \mathbb{E}_\theta[ln p_\theta(X, Z)|X] \\ &= \mathbb{E}_\theta[\sum_{i=1}^{N}\sum_{j=1}^{K} Z_{ik}[ln\pi_k + ln f_k(X_k|\mu_k)] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{K}\mathbb{E}_\theta(Z_{ik}|X_i)[ln\pi_k + ln f_k(X_k|\mu_k)] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{K}\tau_{ik}[ln\pi_k + ln f_k(X_k|\mu_k)]\end{aligned}$$

4.

$$\theta^{q+1} = arg\max_\theta(\mathbb{Q}(\theta^q|\theta)) = arg\max_\theta\{\sum_{i=1}^{N}\sum_{j=1}^{K}\tau_{ik}[ln\pi_k + ln f_k(X_k|\mu_k)]\}$$

5.

Sur l'étape Maximisation nous avons:

$$\theta^{q+1} = arg\max_\theta(\mathbb{Q}(\theta^q|\theta))$$

donc nous avons

$$\frac{\partial \mathbb{Q}(\theta^q|\theta)}{\partial \mu_k} = 0 \leftrightarrow \mu_k^{q+1} = \frac{\sum_{i=1}^N \tau_{ik}^q x_i}{\sum_{i=1}^N \tau_{ik}^q}$$

et

$$\pi_k^{q+1} = \frac{\sum_{i=1}^N \tau_{ik}^q}{N}$$

6.

C'est la terme d'entropie des variables latantes étant donné le Y observé:

$$\begin{aligned} H[p_\theta(Z|X)] &= \sum_i H[p_\theta(Z_i|X_i)] \\ &= -\sum_i \mathbb{E}_\theta[lnP(Z_i = k|X_i)|X_i] \\ &= -\sum_i \sum_k \tau_{ik} ln\tau_{ik} \end{aligned}$$

7.

Nous avons:

$$\mathbb{E}_\theta[lnp_\theta(Z|X)|X] = \mathbb{E}_\theta[lnp_\theta(X,Z) - lnp_\theta(X)|X)]$$

avec $\mathbb{E}_\theta[lnp_\theta(X)|X)] = lnp_\theta(X)$ donc nous aurons:

$$lnp_{\hat{\theta}}(X|\theta = \{\pi, M\}) = \mathbb{E}_\theta[lnp_\theta(X,Z)|X] - \mathbb{E}_\theta[lnp_\theta(Z|X)|X]$$

8.

le critère BIC associé à un modèle à K classes:

$$K_{BIC} = lnp_{\hat{\theta}_K}(X) - \frac{d_K}{2} ln(n)$$

d'où d_K désigne le nombre de paramètres indépendants dans un modèle avec K composants, n le nombre d'échantillons, et

$$lnp_{\hat{\theta}_K}(X) = \sum_{n=1}^N ln\{\sum_{k=1}^K \hat{\pi}_k \times p(X_n|\hat{\mu}_k)\}$$

9.

le critère ICL associé à un modèle à K classes:

$$K_{ICL} = \mathbb{E}_{\hat{\theta}K}[lnp_{\hat{\theta}_K}(X,Z)|X] - \frac{d_K}{2} ln(n)$$

d'où d_K désigne le nombre de paramètres indépendants dans un modèle avec K composants, n le nombre d'échantillons, et

$$\mathbb{E}_{\hat{\theta}K}[lnp_{\hat{\theta}_K}(X,Z)|X] = \sum_{n=1}^N \sum_{k=1}^K \tau_{ik}(ln\pi_k + ln\{_k(X_n))$$

Algorithm 1 EM

Initialize : a random θ^0
$q \leftarrow 0$
while $||\theta^q - \theta^{q+1}|| > \epsilon$ **do**
Expectation step : *compute* $\mathbb{E}_\theta{}^q[lnp_\theta(X,Z)|X]$
Maximazation step : $\theta^{q+1} = \underset{\theta}{\text{argmax}}\,(\mathbb{Q}(\theta^q/\theta)) = \underset{\theta}{\text{argmax}}\mathbb{E}_\theta{}^q[lnp_\theta(X,Z)|X]$
$q \leftarrow q+1$
end while
return θ^q

10.

Exo 3 Programmation de l'algorithme EM

1. E-step

```
E_step <- function(params=list(pi=pi,M=M),X) {
  K <- length(params$pi)
  N <- nrow(X)
  tau <- matrix(NA,N,K)
  for (k in 1:K) {
    for (i in 1:N) {
      tau[i,k]<-(params$pi[k]*prod(dbinom(x = X[i,],size = 1,prob = M[k,])))/sum(sapply(1:K,function(l)
        params$pi[l]*prod(dbinom(x = X[i,],size = 1,prob = M[l,])))
      }))
    }
  }
  return(tau)
}
```

```
params <- list(pi=pi,M=M)
tau <- E_step(params,X)
tau
```

```
##                [,1]         [,2]         [,3]
##   [1,] 1.000000e+00 1.962434e-09 1.343429e-22
##   [2,] 9.999998e-01 1.517554e-07 2.786067e-22
##   [3,] 1.000000e+00 1.436533e-08 5.062065e-18
##   [4,] 1.000000e+00 2.649631e-09 3.262821e-23
##   [5,] 9.999994e-01 5.526113e-07 6.372375e-21
##   [6,] 1.000000e+00 7.596351e-10 8.117695e-22
##   [7,] 1.000000e+00 4.293957e-08 4.222370e-19
##   [8,] 1.000000e+00 9.269419e-09 1.231318e-23
##   [9,] 1.000000e+00 5.644521e-09 1.076182e-20
##  [10,] 1.000000e+00 1.124249e-10 3.123107e-25
##  [11,] 1.000000e+00 6.606581e-09 1.830452e-23
##  [12,] 9.999959e-01 4.144099e-06 1.768969e-18
##  [13,] 9.997607e-01 2.392504e-04 6.826127e-17
##  [14,] 9.999898e-01 1.018837e-05 6.515151e-19
##  [15,] 9.999866e-01 1.339794e-05 2.106336e-19
##  [16,] 1.000000e+00 1.251407e-08 1.440421e-21
##  [17,] 9.999998e-01 2.336489e-07 7.057226e-17
```

```
##  [18,] 1.000000e+00 8.316111e-09 5.318216e-19
##  [19,] 1.000000e+00 1.866297e-10 6.015393e-24
##  [20,] 1.000000e+00 4.247864e-08 4.220191e-20
##  [21,] 9.999790e-01 2.101182e-05 3.500212e-20
##  [22,] 1.000000e+00 2.773787e-12 4.719885e-22
##  [23,] 1.000000e+00 4.864541e-10 4.558773e-24
##  [24,] 1.000000e+00 2.007070e-09 1.122954e-22
##  [25,] 1.000000e+00 2.272822e-08 8.255112e-18
##  [26,] 9.999997e-01 3.393691e-07 1.427195e-22
##  [27,] 9.999347e-01 6.527826e-05 2.994295e-15
##  [28,] 1.000000e+00 1.304608e-09 1.273537e-21
##  [29,] 9.999996e-01 4.287537e-07 3.574623e-23
##  [30,] 9.999976e-01 2.417124e-06 7.106275e-22
##  [31,] 9.999950e-01 4.956616e-06 6.627112e-17
##  [32,] 9.834833e-01 1.651671e-02 3.371908e-18
##  [33,] 9.999765e-01 2.346302e-05 4.236224e-15
##  [34,] 1.000000e+00 4.143584e-12 2.411111e-22
##  [35,] 9.999996e-01 3.979339e-07 1.979177e-18
##  [36,] 9.999999e-01 5.256670e-08 1.261964e-21
##  [37,] 1.000000e+00 2.321484e-11 1.244683e-20
##  [38,] 9.999994e-01 5.826212e-07 1.654667e-18
##  [39,] 1.000000e+00 2.651168e-08 7.168698e-21
##  [40,] 1.000000e+00 1.201200e-12 8.279042e-21
##  [41,] 9.983109e-01 1.689085e-03 4.278928e-13
##  [42,] 1.000000e+00 4.765538e-11 1.456436e-23
##  [43,] 1.000000e+00 2.761589e-09 5.400340e-22
##  [44,] 1.000000e+00 3.081114e-08 1.257340e-25
##  [45,] 9.999347e-01 6.525203e-05 1.122017e-16
##  [46,] 1.000000e+00 1.914305e-14 7.740197e-26
##  [47,] 9.999999e-01 1.054332e-07 4.633910e-22
##  [48,] 9.993658e-01 6.342318e-04 7.117995e-19
##  [49,] 9.987220e-01 1.277984e-03 8.799898e-18
##  [50,] 9.999999e-01 1.083703e-07 9.755649e-22
##  [51,] 1.000000e+00 1.043068e-08 1.499709e-23
##  [52,] 9.999087e-01 9.125336e-05 5.530450e-18
##  [53,] 1.000000e+00 1.361303e-09 1.275677e-21
##  [54,] 1.000000e+00 1.712430e-13 2.086386e-23
##  [55,] 1.000000e+00 2.383685e-09 8.783525e-22
##  [56,] 1.000000e+00 8.832312e-10 5.679735e-23
##  [57,] 9.992131e-01 7.868675e-04 8.722655e-16
##  [58,] 1.000000e+00 2.014178e-10 1.573384e-25
##  [59,] 9.999924e-01 7.585442e-06 5.362845e-21
##  [60,] 1.000000e+00 7.201865e-11 3.356043e-24
##  [61,] 1.000000e+00 1.406890e-09 8.376737e-24
##  [62,] 1.000000e+00 2.051390e-09 5.393299e-24
##  [63,] 1.000000e+00 5.059967e-12 2.559898e-23
##  [64,] 1.000000e+00 3.334703e-11 3.440672e-25
##  [65,] 1.000000e+00 3.881603e-08 2.840836e-21
##  [66,] 1.000000e+00 6.413521e-09 1.048035e-21
##  [67,] 1.000000e+00 1.111901e-10 1.829467e-26
##  [68,] 9.999766e-01 2.340941e-05 9.063327e-19
##  [69,] 9.999401e-01 5.990921e-05 2.527016e-10
##  [70,] 9.999996e-01 3.600922e-07 1.346529e-23
##  [71,] 1.000000e+00 3.032694e-10 1.206735e-22
```

```
##  [72,] 1.000000e+00 3.501100e-11 3.990757e-22
##  [73,] 2.204748e-09 1.000000e+00 7.900886e-12
##  [74,] 9.525034e-07 9.999990e-01 1.905185e-13
##  [75,] 2.892337e-11 1.000000e+00 1.759997e-10
##  [76,] 7.232310e-11 1.000000e+00 1.213128e-10
##  [77,] 6.588361e-07 9.999993e-01 1.018905e-11
##  [78,] 3.319539e-06 9.999967e-01 9.510403e-11
##  [79,] 5.820703e-11 1.000000e+00 3.008391e-08
##  [80,] 1.230421e-06 9.999988e-01 1.572508e-11
##  [81,] 2.175920e-10 1.000000e+00 7.043831e-12
##  [82,] 4.587159e-05 9.999541e-01 2.990934e-10
##  [83,] 8.369584e-08 9.999999e-01 8.973717e-15
##  [84,] 8.335375e-12 1.000000e+00 8.420594e-13
##  [85,] 4.136198e-08 1.000000e+00 1.878342e-10
##  [86,] 1.972701e-09 1.000000e+00 1.690651e-09
##  [87,] 3.854122e-09 1.000000e+00 1.596906e-10
##  [88,] 2.616023e-06 9.999974e-01 9.388763e-16
##  [89,] 5.104537e-09 1.000000e+00 1.547885e-14
##  [90,] 1.037206e-08 1.000000e+00 3.917644e-13
##  [91,] 5.489866e-07 9.999995e-01 8.950823e-14
##  [92,] 7.908986e-07 9.999992e-01 6.743451e-11
##  [93,] 2.280311e-09 1.000000e+00 1.760039e-16
##  [94,] 3.084502e-10 1.000000e+00 3.118172e-12
##  [95,] 8.098203e-07 9.999992e-01 2.065981e-12
##  [96,] 4.216643e-07 9.999996e-01 2.226789e-16
##  [97,] 6.457373e-11 1.000000e+00 9.347510e-11
##  [98,] 2.488065e-04 9.997512e-01 4.849334e-15
##  [99,] 5.897382e-05 9.999410e-01 3.504197e-15
## [100,] 2.319123e-04 9.997681e-01 2.038281e-13
## [101,] 1.011612e-11 1.000000e+00 3.506071e-11
## [102,] 1.239631e-10 1.000000e+00 3.051322e-08
## [103,] 4.824866e-06 9.999952e-01 5.701328e-15
## [104,] 2.320451e-09 1.000000e+00 3.331881e-16
## [105,] 2.598199e-06 9.999857e-01 1.166166e-05
## [106,] 4.355524e-06 9.999956e-01 2.119174e-16
## [107,] 1.047336e-08 1.000000e+00 2.230198e-10
## [108,] 1.440834e-07 9.999999e-01 6.253584e-14
## [109,] 2.438343e-08 1.000000e+00 5.086177e-16
## [110,] 1.132043e-06 9.999989e-01 3.369784e-12
## [111,] 1.924481e-08 1.000000e+00 2.143797e-11
## [112,] 1.663611e-07 9.999998e-01 4.484183e-12
## [113,] 3.913399e-06 9.999961e-01 3.402183e-13
## [114,] 7.887978e-05 9.999211e-01 1.258262e-16
## [115,] 6.154321e-07 9.999994e-01 6.277917e-12
## [116,] 2.098101e-03 9.979019e-01 1.673589e-10
## [117,] 7.667026e-08 9.999999e-01 1.064111e-10
## [118,] 4.560472e-12 9.999998e-01 1.774064e-07
## [119,] 1.708724e-07 9.999998e-01 4.929275e-15
## [120,] 1.238177e-11 9.954116e-01 4.588399e-03
## [121,] 1.628657e-07 9.999998e-01 8.120401e-09
## [122,] 2.283570e-08 1.000000e+00 2.759150e-12
## [123,] 5.394113e-21 2.823241e-16 1.000000e+00
## [124,] 1.272598e-23 1.046361e-10 1.000000e+00
## [125,] 6.394422e-21 1.184582e-10 1.000000e+00
```

```
## [126,] 8.891732e-18 3.961783e-06 9.999960e-01
## [127,] 1.145882e-23 2.597865e-12 1.000000e+00
## [128,] 3.411265e-22 9.638822e-11 1.000000e+00
## [129,] 2.900413e-19 2.635289e-10 1.000000e+00
## [130,] 6.670464e-18 3.149848e-09 1.000000e+00
## [131,] 1.085858e-22 7.309034e-11 1.000000e+00
## [132,] 2.148470e-21 5.306475e-15 1.000000e+00
## [133,] 4.782513e-22 1.256391e-12 1.000000e+00
## [134,] 1.473548e-24 1.144497e-13 1.000000e+00
## [135,] 1.725093e-20 1.684375e-08 1.000000e+00
## [136,] 1.794804e-21 8.765568e-10 1.000000e+00
## [137,] 2.707713e-22 4.167593e-14 1.000000e+00
## [138,] 1.406167e-20 1.983226e-13 1.000000e+00
## [139,] 1.790542e-21 2.327372e-10 1.000000e+00
## [140,] 7.330809e-22 1.577023e-13 1.000000e+00
## [141,] 3.625116e-19 8.393249e-13 1.000000e+00
## [142,] 3.347873e-18 1.413918e-11 1.000000e+00
## [143,] 8.814374e-25 5.705053e-15 1.000000e+00
## [144,] 1.886990e-20 4.345570e-10 1.000000e+00
## [145,] 1.319593e-22 4.205325e-15 1.000000e+00
## [146,] 8.723105e-24 1.252641e-12 1.000000e+00
## [147,] 2.188405e-16 3.500272e-11 1.000000e+00
## [148,] 2.477997e-25 5.258181e-11 1.000000e+00
## [149,] 1.120399e-21 1.544892e-09 1.000000e+00
## [150,] 1.116678e-21 5.128470e-12 1.000000e+00
## [151,] 1.898214e-20 7.905784e-12 1.000000e+00
## [152,] 3.567632e-18 4.038377e-08 1.000000e+00
## [153,] 6.317362e-24 1.040781e-13 1.000000e+00
## [154,] 2.134816e-24 5.132762e-14 1.000000e+00
## [155,] 1.761080e-20 3.789363e-13 1.000000e+00
## [156,] 2.426744e-14 1.259393e-08 1.000000e+00
## [157,] 2.939017e-21 5.344971e-12 1.000000e+00
## [158,] 8.804453e-22 3.951594e-12 1.000000e+00
## [159,] 1.280959e-24 1.401037e-11 1.000000e+00
## [160,] 8.016581e-23 7.001784e-14 1.000000e+00
## [161,] 7.296878e-24 2.715356e-12 1.000000e+00
## [162,] 2.145221e-20 7.437780e-08 9.999999e-01
## [163,] 2.386324e-18 8.418212e-11 1.000000e+00
## [164,] 1.476439e-22 1.455357e-14 1.000000e+00
## [165,] 1.124454e-21 1.417273e-17 1.000000e+00
## [166,] 3.200650e-19 4.585119e-08 1.000000e+00
## [167,] 4.673780e-20 2.032189e-10 1.000000e+00
## [168,] 6.224285e-23 3.989360e-13 1.000000e+00
## [169,] 1.978092e-23 2.258079e-11 1.000000e+00
## [170,] 1.368542e-25 1.913850e-15 1.000000e+00
## [171,] 1.737845e-21 7.361338e-09 1.000000e+00
## [172,] 4.379555e-20 2.957845e-11 1.000000e+00
## [173,] 1.093427e-21 1.290984e-10 1.000000e+00
## [174,] 3.142467e-22 1.959519e-14 1.000000e+00
## [175,] 2.093844e-26 5.802907e-13 1.000000e+00
## [176,] 1.916271e-24 3.499837e-13 1.000000e+00
## [177,] 1.522468e-18 8.617126e-08 9.999999e-01
## [178,] 2.837738e-20 1.656075e-09 1.000000e+00
## [179,] 1.517615e-18 1.687910e-07 9.999998e-01
```

```
## [180,] 1.513109e-14 2.303515e-05 9.999770e-01
## [181,] 5.966268e-21 6.751405e-09 1.000000e+00
## [182,] 5.455970e-25 3.307447e-12 1.000000e+00
## [183,] 1.732577e-20 8.778167e-12 1.000000e+00
## [184,] 1.949386e-21 8.785107e-13 1.000000e+00
## [185,] 3.486490e-20 4.615990e-11 1.000000e+00
## [186,] 4.514900e-22 2.245164e-11 1.000000e+00
## [187,] 1.070819e-18 1.046364e-07 9.999999e-01
## [188,] 2.277335e-21 3.083005e-10 1.000000e+00
## [189,] 1.119846e-25 3.195095e-15 1.000000e+00
## [190,] 2.508430e-21 5.926928e-08 9.999999e-01
## [191,] 2.270271e-24 5.812815e-12 1.000000e+00
## [192,] 2.761580e-25 2.506729e-12 1.000000e+00
## [193,] 8.298248e-24 6.719293e-11 1.000000e+00
## [194,] 6.039586e-13 3.043075e-07 9.999997e-01
## [195,] 6.873376e-20 1.944737e-05 9.999806e-01
## [196,] 9.177894e-23 1.825801e-12 1.000000e+00
## [197,] 1.321044e-22 1.253377e-11 1.000000e+00
## [198,] 3.389250e-24 3.358766e-14 1.000000e+00
## [199,] 6.993902e-26 1.958957e-12 1.000000e+00
## [200,] 4.846161e-25 2.185752e-13 1.000000e+00
```

Selon les résultats des variables τ_{ik}, nous avons constaté que de quel composante de K la variable X_i appartient, la valeur correspondante de τ_{ik} est très proche de 1, et la valeur de τ_{ik} pour laquelle k n'est pas égal à K est très petite.

2. M-step

```
M_step <- function(X, tau, params) {
  N <- nrow(X)
  K <- ncol(tau)
  pi <- params$pi
  mu <- params$M

  for (k in 1:K) {
    N_k <- sum(tau[,k])
    pi[k] <- N_k/N

    if(N_k!=0) {
      mu[k,] <- 1/N_k*rowSums(sapply(1:N, function(n){
        tau[n,k]*X[n,]
      }))
    }
  }

  return(list(pi=pi,M=mu))
}
```

```
params <- M_step(X, tau, params)
matrix(params$pi,1,3,dimnames = list(c("val"), c("pi_1","pi_2","pi_3")))
```

```
##          pi_1     pi_2      pi_3
## val 0.3599063 0.250071 0.3900228
```

```r
t(params$M)
```

```
##               [,1]       [,2]       [,3]
##  [1,] 1.525833e-01 0.80001906 0.80770353
##  [2,] 2.776566e-01 0.60001673 0.65386671
##  [3,] 1.528087e-01 0.21995074 0.84610415
##  [4,] 6.804702e-01 0.52013885 0.24363425
##  [5,] 3.611824e-01 0.53987969 0.42305208
##  [6,] 8.612957e-01 0.10002855 0.12819779
##  [7,] 8.752101e-01 0.29994050 0.12819765
##  [8,] 3.260523e-06 0.21984137 0.97436077
##  [9,] 2.220336e-01 0.90000835 0.84616283
## [10,] 9.444424e-01 0.56010631 0.15389594
## [11,] 3.750674e-01 0.31986119 0.57694808
## [12,] 5.417709e-01 0.23989311 0.57694822
## [13,] 8.749596e-01 0.14034692 0.14101711
## [14,] 2.083595e-01 0.19998459 0.80764456
## [15,] 2.220035e-01 0.60013702 0.88462197
## [16,] 2.637270e-01 0.06031523 0.70508679
## [17,] 2.220809e-01 0.75997978 0.73078504
## [18,] 2.500314e-01 0.79973040 0.80770359
## [19,] 8.194206e-01 0.82010903 0.11537773
## [20,] 3.053861e-01 0.18030755 0.74354623
## [21,] 8.055189e-01 0.04034334 0.20511623
## [22,] 9.727944e-02 0.73974501 0.84610385
## [23,] 6.943934e-01 0.58007720 0.28209374
## [24,] 9.723028e-02 0.63975218 0.96154070
## [25,] 9.444333e-01 0.97999993 0.08979732
## [26,] 6.251361e-01 0.12000452 0.39741255
## [27,] 6.943928e-01 0.70004468 0.17953527
## [28,] 6.804465e-01 0.28033305 0.35895311
## [29,] 9.444009e-01 0.78010446 0.03851752
## [30,] 5.140154e-01 0.55985230 0.48715060
## [31,] 2.916235e-05 0.19981047 0.99999985
## [32,] 8.333135e-01 0.76003342 0.11543670
## [33,] 3.889468e-01 0.25998914 0.69226698
## [34,] 6.110065e-01 0.54022682 0.38459268
## [35,] 2.776223e-01 0.60015826 0.73072613
## [36,] 3.195127e-01 0.04001065 0.57688905
## [37,] 4.167723e-02 0.25983582 0.89744173
## [38,] 5.559882e-02 0.19981100 0.93590103
## [39,] 9.027312e-01 0.70020660 0.05127921
## [40,] 5.834696e-01 0.35982911 0.37183170
## [41,] 6.249077e-01 0.88002598 0.28209339
## [42,] 2.500226e-01 0.57980556 0.87180236
## [43,] 6.943930e-01 0.48019833 0.33331378
## [44,] 3.473345e-01 0.37976954 0.75642457
## [45,] 3.334190e-01 0.11987588 0.62822722
## [46,] 6.251275e-01 0.04003916 0.44869190
## [47,] 5.554707e-01 0.64005724 0.34619231
## [48,] 3.750864e-01 0.71972040 0.70514572
## [49,] 8.194023e-01 0.28028831 0.14101741
## [50,] 8.333248e-01 0.62014861 0.15383727
```

Ce sont les valeurs du paramètre calculée à l'aide de la fonction M_step écrite par nous-mêmes. Après comparaison avec les valeur réelles, nous constatons que les résultats sont plus précis.

3. l'algorithme EM

```
Init.EM <- function(X, K=3) {
  pi <- rep(x = 1/K, times = K)
  # pi <- c(0.9,0.05,0.05)
  M <- matrix(0.5,K,ncol(X))
  params <- list(pi = pi, M = M)
  return(params)
}

EM <- function(X, K) {
  params <- Init.EM(X, K)
  iter <- 0
  params.new <- params
  repeat{
    tau <- E_step(params = params.new, X)
    params <- M_step(X, tau, params)
    if((sum(unlist(params.new) - unlist(params))^2) / sum(unlist(params.new))^2 < 1e-20) break
    params.new <- params
  }
  return(list(params = params.new, tau = tau))
}
```

```
matrix(EM(X,3)$params$pi,1,3,dimnames = list(c("val"), c("pi_1","pi_2","pi_3")))
```

```
##         pi_1      pi_2      pi_3
## val 0.359945 0.2500194 0.3900356
```

```
t(EM(X,3)$params$M)
```

```
##                [,1]       [,2]       [,3]
##  [1,] 1.526415e-01 0.80002521 0.80770988
##  [2,] 2.777123e-01 0.59996592 0.65387801
##  [3,] 1.527974e-01 0.21998861 0.84607634
##  [4,] 6.804819e-01 0.52007255 0.24365908
##  [5,] 3.611545e-01 0.53997510 0.42303831
##  [6,] 8.612178e-01 0.10002784 0.12819365
##  [7,] 8.751455e-01 0.29995980 0.12819341
##  [8,] 4.693092e-06 0.21983335 0.97436151
##  [9,] 2.221125e-01 0.89999455 0.84616790
## [10,] 9.444432e-01 0.56002343 0.15392359
## [11,] 3.750365e-01 0.31986238 0.57696193
## [12,] 5.417251e-01 0.23987375 0.57696188
## [13,] 8.749691e-01 0.14022655 0.14101256
## [14,] 2.083489e-01 0.20000851 0.80761812
## [15,] 2.220671e-01 0.60008358 0.88462569
## [16,] 2.637684e-01 0.06022721 0.70506361
## [17,] 2.221403e-01 0.75996556 0.73079376
## [18,] 2.500213e-01 0.79981975 0.80770991
## [19,] 8.194305e-01 0.82013677 0.11537412
## [20,] 3.054286e-01 0.18023601 0.74352187
```

```
## [21,] 8.055093e-01 0.04024074 0.20510946
## [22,] 9.728407e-02 0.73987560 0.84607615
## [23,] 6.944170e-01 0.58000424 0.28211724
## [24,] 9.725037e-02 0.63978872 0.96154197
## [25,] 9.444377e-01 0.97999849 0.08982715
## [26,] 6.250776e-01 0.12001652 0.39739962
## [27,] 6.944162e-01 0.69999674 0.17956214
## [28,] 6.804600e-01 0.28026586 0.35894137
## [29,] 9.443936e-01 0.78007845 0.03854917
## [30,] 5.139902e-01 0.55992402 0.48713482
## [31,] 4.194092e-05 0.19978193 0.99999977
## [32,] 8.333237e-01 0.75999554 0.11546560
## [33,] 3.889184e-01 0.26002331 0.69224424
## [34,] 6.110442e-01 0.54018927 0.38458008
## [35,] 2.776670e-01 0.60017435 0.73070231
## [36,] 3.195046e-01 0.03998097 0.57687014
## [37,] 4.167287e-02 0.25983790 0.89744503
## [38,] 5.560624e-02 0.19978158 0.93590306
## [39,] 9.027233e-01 0.70022243 0.05127760
## [40,] 5.834172e-01 0.35983728 0.37185223
## [41,] 6.249464e-01 0.88000402 0.28211688
## [42,] 2.500120e-01 0.57985020 0.87180659
## [43,] 6.944171e-01 0.48015488 0.33330295
## [44,] 3.473122e-01 0.37977503 0.75643244
## [45,] 3.333844e-01 0.11984788 0.62823921
## [46,] 6.250711e-01 0.04003175 0.44867718
## [47,] 5.555053e-01 0.64000212 0.34621378
## [48,] 3.750551e-01 0.71980449 0.70515534
## [49,] 8.194161e-01 0.28019946 0.14101275
## [50,] 8.333320e-01 0.62013692 0.15383229
```

4. l'évolution de la vraisemblance à chaque demi-étape

La vraisemblance est précisement la vraisemblance complète:

$$\mathcal{L} = p(X, Z|\theta = \{\pi, M\}) = \prod_n \prod_k (\pi_k \times \mathcal{B}(X_n|\mu_k))^{Z_{nk}}$$

Mais étant donné que la probabilité et la valeur de la proportion sont toutes deux inférieures à 1, leur produit tend vers 0 après plusieurs fois de multiplications. Nous calculons ici donc la log-vraisemblance:

$$ln\mathcal{L} = lnp(X, Z|\theta = \{\pi, M\}) = \sum_n \sum_k Z_{nk}(ln\pi_k + ln\{_k(X_n))$$

```
log_vrsblc <- function(X, params, tau) {
  N <- nrow(X)
  K <- length(params$pi)
  # Z <- rep(1:K,params$pi*N)
  Z <- apply(tau, 1, which.max) # uncertain
  logL <- 0

  for (n in 1:N) {
    for (k in 1:K) {
      logL <- logL + ifelse(Z[n]==k,1,0)*(log(params$pi[k])+log(prod(dbinom(X[n,],1,params$M[k,]))))
```

```
    }
  }

  return(logL)
}
```

```
log_vrsblc(X,params,tau)
```

```
## [1] -5236.743
```

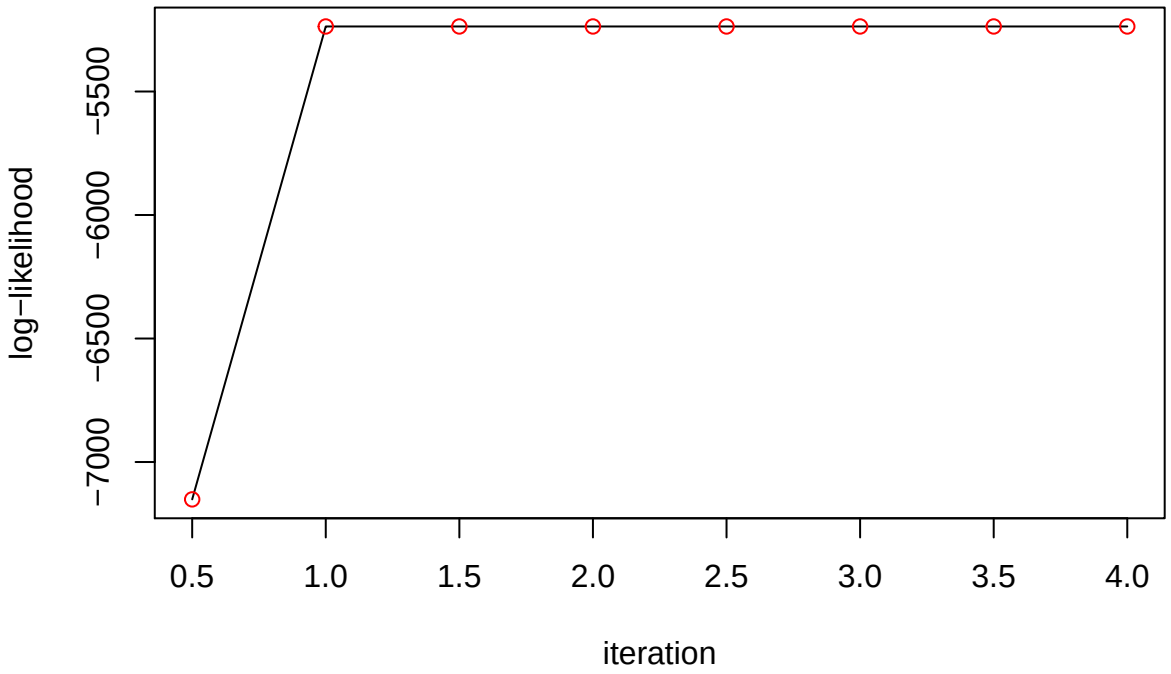
```
EM.trace <- function(X, K) {
  params <- Init.EM(X, K)
  iter <- 0
  params.new <- params
  logLs <- c()

  repeat{
    tau <- E_step(params = params.new, X)
    logLs <- append(logLs, log_vrsblc(X,params,tau))  # calcule log-vraisemblance après l'étape E

    params <- M_step(X, tau, params)
    logLs <- append(logLs, log_vrsblc(X,params,tau))  # calcule log-vraisemblance après l'étape M

    if((sum(unlist(params.new) - unlist(params))^2) / sum(unlist(params.new))^2 < 1e-20) break
    params.new <- params
  }
  return(list(params = params.new, tau = tau, logLs = logLs))
}
```

```
logLs <- EM.trace(X,3)$logLs
plot(x = seq(from = 0.5, to = length(logLs)/2, by = 1/2), y = logLs,
     type = "l", xlab = "iteration", ylab = "log-likelihood")
points(x = seq(from = 0.5, to = length(logLs)/2, by = 1/2), y = logLs, col = "red")
```



5. La fonction BIC

```
# la fonction qui prend la sortie de l'algorithme EM et rend le critère BIC.
BIC <- function(params, X) {
  N <- nrow(X)
  logp <- 0
  for (n in 1:N) {
    p <- 0
    for (k in 1:K) {
      p <- p + params$pi[k]*prod(dbinom(x=X[n,],size=1,prob=params$M[k,]))
    }
    logp <- logp + log(p)
  }
  d_k <- 1
  return(logp - d_k/2*log(N))
}
```

```
params <- EM(X, 3)$params
BIC(params, X)
```

```
## [1] -5239.349
```

6. La fonction ICL

```
# la fonction qui prend la sortie de l'algorithme EM et rend le critère ICL.
ICL <- function(params, tau, X) {
  N <- nrow(X)
  K <- length(params$pi)
  rtn <- 0
  for (n in 1:N) {
    for (k in 1:K) {
      rtn <- rtn + tau[n,k]*(log(params$pi[k])+sum(dbinom(x=X[n,],size=1,prob=params$M[Z[n],],log=T)))
    }
  }
  d_k <- 1
  return(rtn - d_k/2*log(N))
}
```

```
params <- EM(X, 3)$params
tau <- EM(X, 3)$tau
ICL(params, tau, X)
```

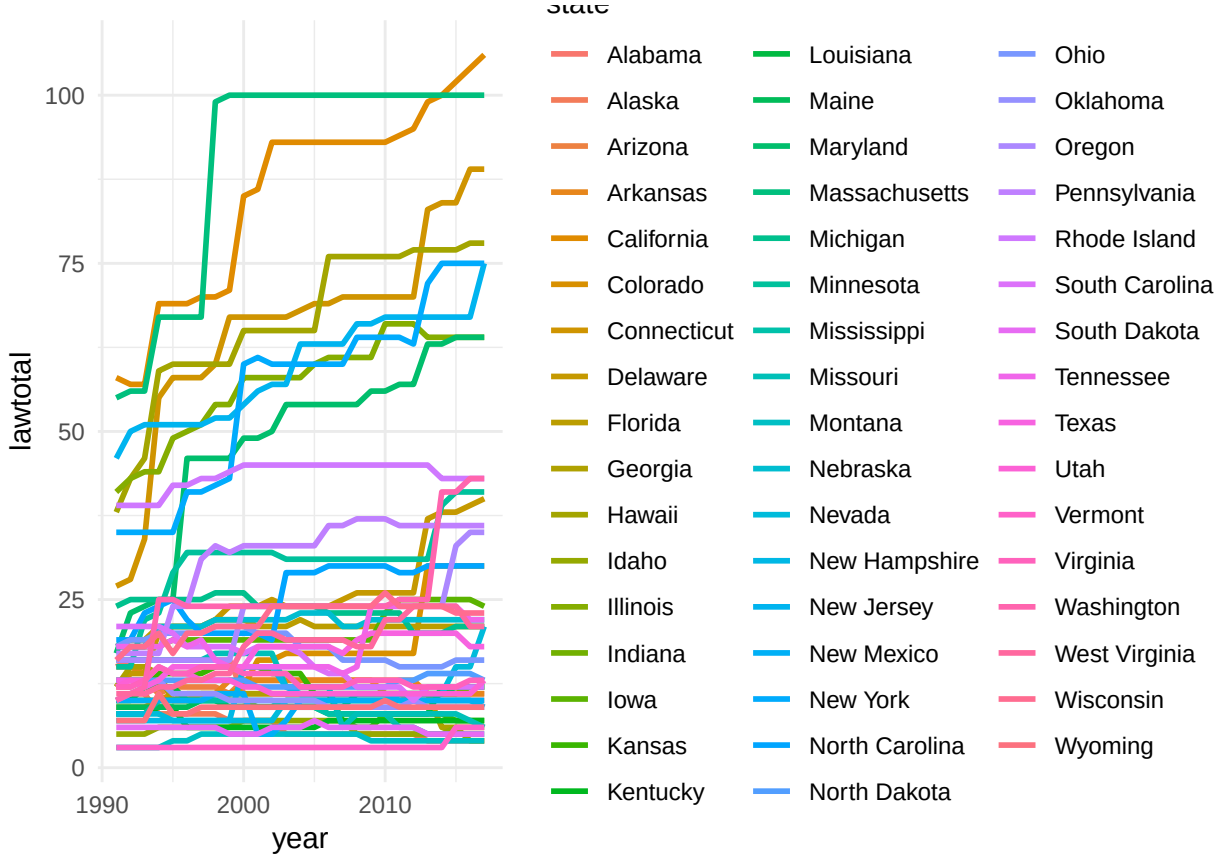
```
## [1] -5239.394
```

Exo 4 Données state-firearms

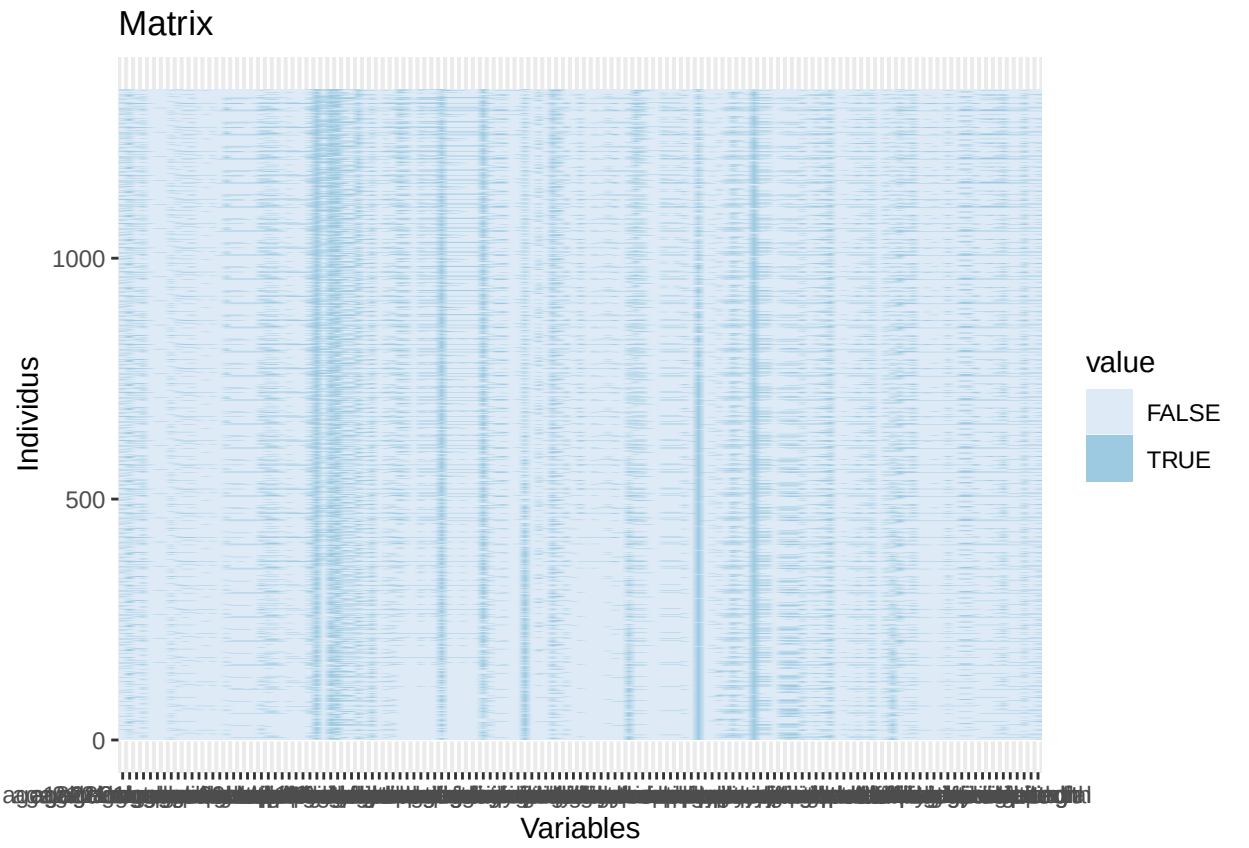
```
dat <- read.csv(file = "raw_data.csv")
rows <- as.matrix(dat[,-which(colnames(dat)%in%c("state","year","lawtotal"))])
cols <- t(rows)
```

```
ggplot(dat) +
 aes(x = year, y = lawtotal, colour = state) +
```

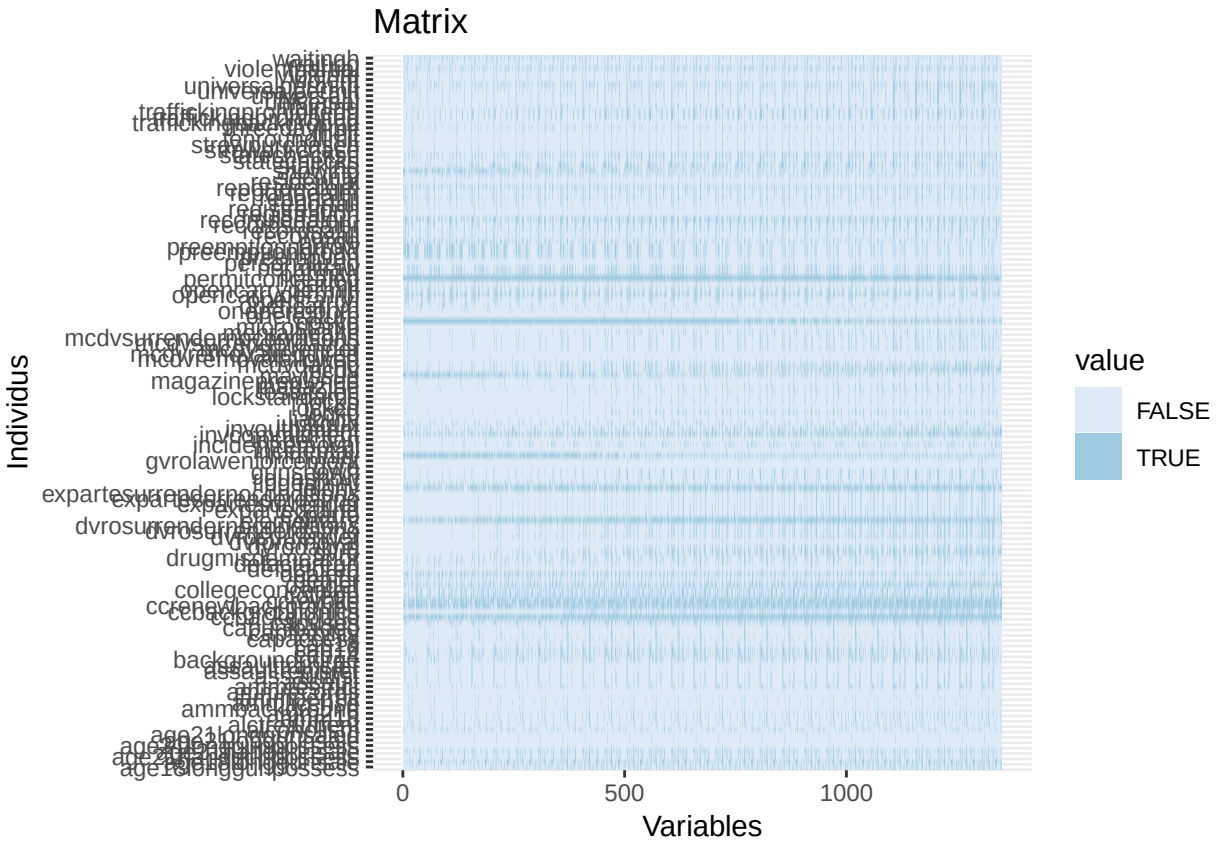
```
  geom_line(size = 1L) +
  scale_color_hue() +
  theme_minimal()
```




```
melt_rows <- melt(rows)
melt_rows$value <- as.logical(melt_rows$value)
ggplot(melt_rows, aes(x = Var2, y = Var1)) +
  geom_raster(aes(fill=value)) +
  scale_fill_brewer(aesthetics = "fill") +
  labs(x="Variables", y="Individus", title="Matrix")
```




```
melt_cols <- melt(cols)
melt_cols$value <- as.logical(melt_cols$value)
ggplot(melt_cols, aes(x = Var2, y = Var1)) +
  geom_raster(aes(fill=value)) +
  scale_fill_brewer(aesthetics = "fill") +
  labs(x="Variables", y="Individus", title="Matrix")
```



Ici selon le première plot nous pouvons voir que dans le plupart des états, le nombre de lois relatives aux munitions augmente avec le temps .

Après nous utilisons BIC pour determiner le nombre de K respectivement sur l'analise de lignes et de colonnes:

```
library(mclust)
```

```
## Package 'mclust' version 5.4.3
## Type 'citation("mclust")' for citing this R package in publications.
```

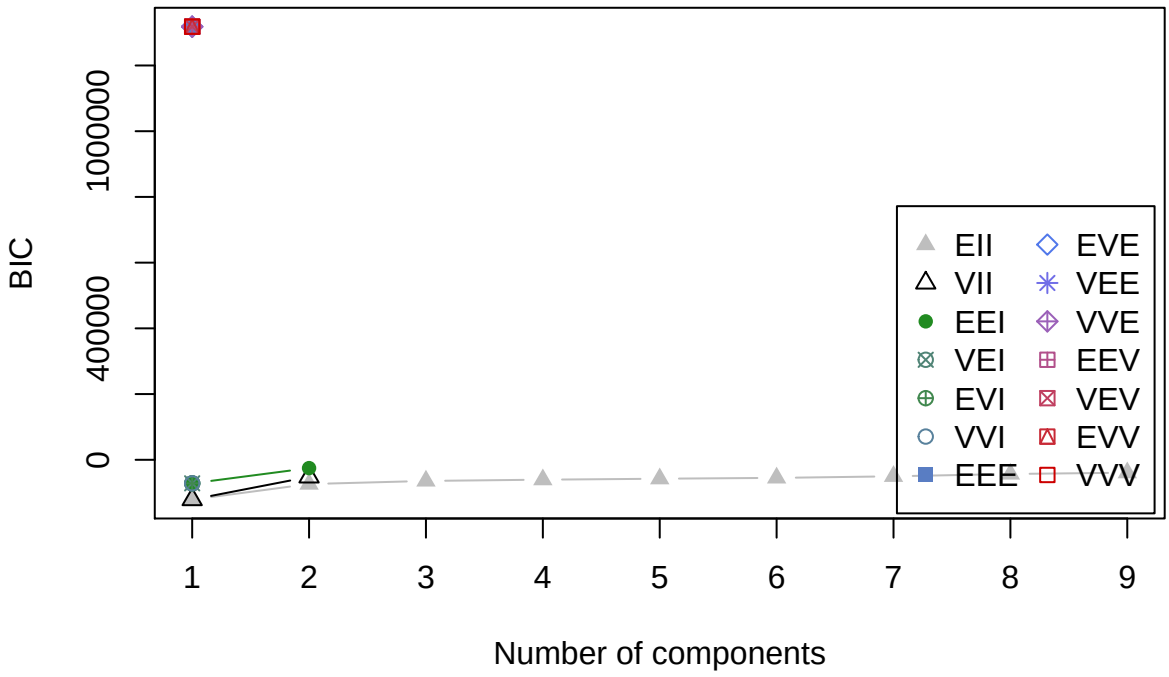
```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
clusters_mclust <- Mclust(rows)
print(clusters_mclust$G)
```

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

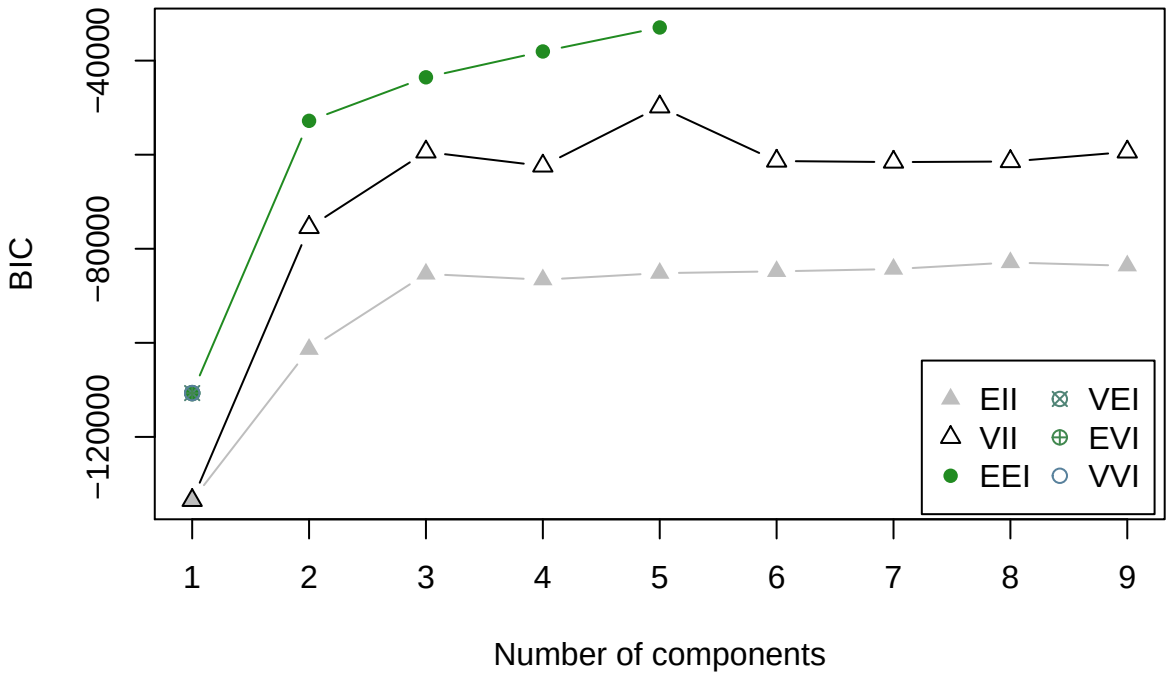
```
plot(clusters_mclust, what = "BIC")
```




```
clusters_mclust <- Mclust(cols)
print(clusters_mclust$G)
```

```
## [1] 5
```

```
plot(clusters_mclust, what = "BIC")
```



Nous supposons: pour les lignes $K = 2$ et les colonnes $k = 5$

```
res_rows <- EM(X = rows, K = 2)
print(res_rows$params)
```

```
## $pi
## [1] 0.004687295 0.995312705
##
## $M
##              [,1]      [,2]         [,3]         [,4]         [,5]
## [1,] 0.001903206 0.9974205 0.0001883418 0.0001969313 2.821493e-12
## [2,] 0.200188682 0.3398809 0.1458680288 0.2485716128 1.488458e-02
##               [,6]         [,7]         [,8]       [,9]      [,10]
## [1,] 0.0000064003 0.0000064003 3.694814e-06 0.99723433 0.99740946
## [2,] 0.0178614698 0.0178614698 1.414035e-01 0.09279768 0.09949491
##              [,11]     [,12]      [,13]     [,14]        [,15]     [,16]
## [1,] 0.0001822589 0.9972286 0.99911347 0.9972286 2.669788e-13 0.9972491
## [2,] 0.1175873493 0.0392132 0.03325049 0.0392132 5.209604e-03 0.1448936
##              [,17]        [,18]        [,19]        [,20]     [,21]
## [1,] 7.996901e-07 7.996910e-07 7.932517e-07 7.895849e-07 0.9999126
## [2,] 9.005172e-02 9.079595e-02 6.177102e-02 4.614220e-02 0.2282348
##              [,22]        [,23]        [,24]       [,25]        [,26]
## [1,] 0.0001956286 0.0001941143 5.502391e-06 0.000188133 0.0001816037
## [2,] 0.2917369103 0.2336870428 8.781902e-02 0.112377718 0.0565605611
##              [,27]        [,28]     [,29]       [,30]     [,31]       [,32]
## [1,] 0.0001881267 0.0001956286 0.9999820 0.001875239 0.9993362 0.002183646
## [2,] 0.0535836137 0.2917369103 0.8385024 0.203165730 0.6963576 0.671284421
##          [,33]     [,34]     [,35]     [,36]     [,37]      [,38]
## [1,] 0.9978695 0.9978664 0.9993181 0.9991231 0.9999382 0.99723351
## [2,] 0.3614614 0.2840616 0.4135506 0.1352098 0.3287056 0.03623626
##          [,39]     [,40]     [,41]     [,42]      [,43]     [,44]
## [1,] 0.9991046 0.9974445 0.6105926 0.6085180 0.60851021 0.6104054
## [2,] 0.1500945 0.1813600 0.2851412 0.1697954 0.02243809 0.1839269
##          [,45]     [,46]     [,47]     [,48]      [,49]      [,50]
## [1,] 0.6085131 0.6085190 0.9999795 0.6104044 0.60851272 0.61040441
## [2,] 0.1229090 0.1601204 0.7432410 0.1191790 0.08793025 0.09908478
##           [,51]      [,52]     [,53]     [,54]      [,55]      [,56]
## [1,] 0.60851271 0.60851804 0.9993198 0.9978601 0.00187549 0.00205882
## [2,] 0.06783606 0.08123216 0.6784962 0.1441465 0.11088131 0.14660345
##            [,57]       [,58]     [,59]        [,60]       [,61]
## [1,] 0.001312387 0.001313207 0.9980490 4.966456e-09 0.000635692
## [2,] 0.002970736 0.026041836 0.5050968 1.041921e-02 0.180100464
##              [,62]     [,63]       [,64]        [,65]        [,66]
## [1,] 1.801949e-09 0.9993314 0.001903854 0.0001897368 2.406418e-06
## [2,] 3.497877e-02 0.4522505 0.216561720 0.1235411474 3.125761e-02
##              [,67]        [,68]        [,69]        [,70]        [,71]
## [1,] 2.070449e-05 2.821493e-12 2.192547e-08 5.146240e-09 6.953182e-06
## [2,] 1.361938e-01 1.488458e-02 5.507296e-02 4.018837e-02 8.781901e-02
##              [,72]        [,73]     [,74]       [,75]       [,76]
## [1,] 0.0001816255 4.990315e-09 0.9974216 0.002089426 0.002086311
## [2,] 0.0870739566 2.455956e-02 0.3398809 0.290239533 0.209862799
##            [,77]        [,78]        [,79]        [,80]        [,81]
## [1,] 0.001887494 1.416709e-10 8.310033e-06 8.225650e-06 6.920614e-06
## [2,] 0.034969882 2.976917e-03 1.146113e-01 8.930746e-02 1.049363e-01
##            [,82]        [,83]     [,84]        [,85]        [,86]
## [1,] 0.001876042 1.858835e-05 0.9999813 3.794052e-11 0.0001821755
## [2,] 0.118323602 2.307102e-02 0.8109659 2.083842e-02 0.0602817042
##              [,87]     [,88]     [,89]     [,90]     [,91]     [,92]
## [1,] 1.863449e-07 0.9972360 0.9980533 0.9978635 0.9972352 0.9999888
```

```
## [2,] 1.004709e-01 0.1203341 0.3778336 0.1947541 0.1009842 0.9456713
##          [,93]     [,94]        [,95]     [,96]     [,97]     [,98]
## [1,] 0.9999133 0.9999127 1.859113e-05 0.9972344 0.9972377 0.9972364
## [2,] 0.2706558 0.2431194 2.679216e-02 0.1798725 0.2319685 0.2215493
##          [,99]       [,100]      [,101]    [,102]    [,103]       [,104]
## [1,] 0.9993037 7.197018e-06 0.002058677 0.9972360 0.9993092 0.0000064003
## [2,] 0.2215396 8.633055e-02 0.180093762 0.1999667 0.4061084 0.0178614698
##           [,105]     [,106]    [,107]    [,108]    [,109]     [,110]
## [1,] 6.729124e-06 0.99723434 0.9992861 0.9972361 0.9993054 0.99722710
## [2,] 6.028253e-02 0.07568041 0.1642340 0.1173572 0.2594953 0.01539788
##         [,111]       [,112]    [,113]    [,114]    [,115]     [,116]
## [1,] 0.9972304 0.0001940372 0.9157822 0.9990979 0.9992812 0.99722770
## [2,] 0.1709418 0.2850388553 0.2784943 0.1515830 0.2609838 0.03325937
##         [,117]       [,118]     [,119]    [,120]     [,121]    [,122]
## [1,] 0.9972280 6.477240e-09 0.99722770 0.9999127 0.99722770 0.9972366
## [2,] 0.0518651 4.911912e-02 0.03325937 0.1530676 0.03325937 0.2103859
##         [,123]       [,124]      [,125]      [,126]    [,127]    [,128]
## [1,] 0.9972373 3.117794e-06 0.001873985 0.002057315 0.9991035 0.9999151
## [2,] 0.2334570 1.146113e-01 0.071437174 0.107159311 0.1359542 0.2639578
##           [,129]       [,130]       [,131]       [,132]       [,133]
## [1,] 7.571194e-06 7.646222e-06 1.641867e-05 8.734201e-06 0.0008185765
## [2,] 6.921328e-02 7.516711e-02 2.820628e-01 8.037671e-02 0.1905188105
```

```
# rownames(cols) <- 1:nrow(cols)
# res_cols <- EM(X = cols, K = 5) # out of bounds
```