

TP6: Classification non supervisée sur modèles de mélanges : Algorithme EM et CEM

DJEBALI Wissam

3 mars 2018

Algorithme EM : Expectation-Maximisation (Maximum de Vraisemblance)

Packages R : Rcpp, Rmixmod

Lorsque les seules données dont on dispose ne permettent pas l'estimation des paramètres, et/ou que l'expression de la vraisemblance est analytiquement impossible à maximiser, l'algorithme EM peut être une solution tout comme une façon d'estimer les données cachées ou manquantes.

L'algorithme EM tire son nom du fait qu'à chaque itération il opère deux étapes distinctes :

- la phase « Expectation », souvent désignée comme « l'étape E », procède comme son nom le laisse supposer à l'estimation des données inconnues, sachant les données observées et la valeur des paramètres déterminée à l'itération précédente ;
- la phase « Maximisation », ou « étape M », procède donc à la maximisation de la vraisemblance, rendue désormais possible en utilisant l'estimation des données inconnues effectuée à l'étape précédente, et met à jour la valeur du ou des paramètre(s) pour la prochaine itération.

L'algorithme EM avec notations

L'algorithme EM vise à maximiser la vraisemblance du mélange dans un contexte non supervisé.

- 1) Déterminer une situation initiale
- 2) **Estimation** des probabilités conditionnelles (probabilités a posteriori)

$$t_{ik} = \frac{p_k \phi_k(x_i; \alpha_k)}{\sum_{l=1}^s p_l \phi_l(x_i; \alpha_l)}$$

que x_i

- 3) **Maximisation** : calcul des paramètres du mélange

$$p_k = \frac{\sum_{i=1}^n t_{ik}}{n}$$

α_k = dépendant du modèle

- 4) Itérer les étapes 2 et 3, jusqu'à la convergence (évolution très faible de L)

En bref, l'algorithme EM procède selon un mécanisme extrêmement naturel : s'il existe un obstacle pour appliquer la méthode MV, on fait simplement sauter cet obstacle puis on applique effectivement cette méthode.

```
data("geyser")
gey<-geyser

# Plusieurs modèles et plusieurs critères
modu <- mixmodCluster(gey,nbCluster=2:5,criterion=c("ICL","BIC"),
                      strategy=mixmodStrategy('EM'), models=mixmodGaussianModel())
modu
```

```

## *****
## *** INPUT:
## *****
## * nbCluster = 2 3 4 5
## * criterion = ICL BIC
## *****
## *** MIXMOD Models:
## * list = Gaussian_pk_L_I Gaussian_pk_Lk_I Gaussian_pk_L_B Gaussian_pk_Lk_B Gaussian_pk_L_Bk Gaussian
## * This list includes models with free and equal proportions.
## *****
## * data (limited to a 10x10 matrix) =
##      Duration Waiting.Time
## [1,] 3.6      79
## [2,] 1.8      54
## [3,] 3.333    74
## [4,] 2.283    62
## [5,] 4.533    85
## [6,] 2.883    55
## [7,] 4.7      88
## [8,] 3.6      85
## [9,] 1.95     51
## [10,] 4.35    85
## * ... ...
## *****
## *** MIXMOD Strategy:
## * algorithm      = EM
## * number of tries = 1
## * number of iterations = 200
## * epsilon        = 0.001
## *** Initialization strategy:
## * algorithm      = smallEM
## * number of tries = 10
## * number of iterations = 5
## * epsilon        = 0.001
## * seed           = NULL
## *****
##
##
## *****
## *** BEST MODEL OUTPUT:
## *** According to the ICL criterion
## *****
## * nbCluster = 2
## * model name = Gaussian_pk_Lk_D_Ak_D
## * criterion = ICL(2321.3703) BIC(2320.2833)
## * likelihood = -1132.1126
## *****
## *** Cluster 1
## * proportion = 0.6432
## * means      = 4.2915 79.9893
## * variances = | 0.1588 0.6810 |
##               | 0.6810 35.7667 |
## *** Cluster 2
## * proportion = 0.3568

```

```
## * means      = 2.0387 54.5041
## * variances  = |      0.0783      0.6467 |
##              |      0.6467     33.8930 |
## *****

# mixmodCluster affiche par défaut en premier le résultat de ICL car
# il apparait en premier dans le vecteur

# Résumé des résultats
summary(modu)

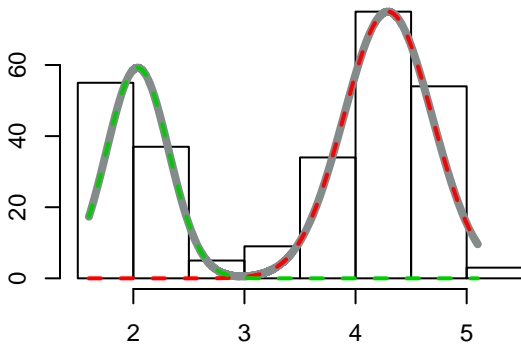
## *****
## * Number of samples      = 272
## * Problem dimension      = 2
## *****
## *      Number of cluster = 2
## *      Model Type = Gaussian_pk_Lk_D_Ak_D
## *      Criterion = ICL(2321.3703) BIC(2320.2833)
## *      Parameters = list by cluster
## *      Cluster 1 :
##              Proportion = 0.6432
##              Means = 4.2915 79.9893
##              Variances = |      0.1588      0.6810 |
##                          |      0.6810     35.7667 |
## *      Cluster 2 :
##              Proportion = 0.3568
##              Means = 2.0387 54.5041
##              Variances = |      0.0783      0.6467 |
##                          |      0.6467     33.8930 |
## *      Log-likelihood = -1132.1126
## *****

# Classification des individus
modu['partition']

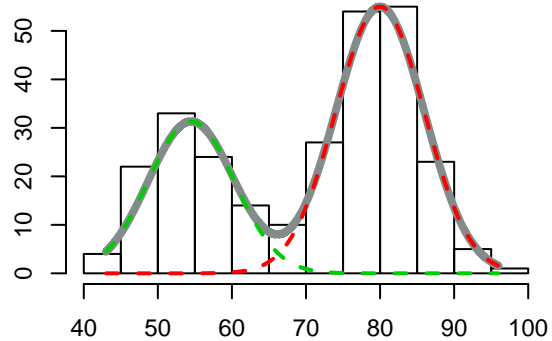
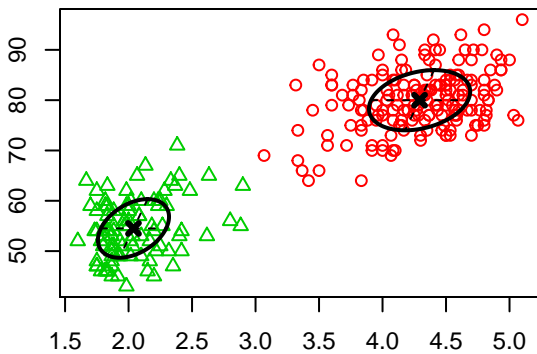
## [1] 1 2 1 2 1 2 1 1 2 1 2 1 1 2 1 2 2 1 2 2 1 1 1 2 1 1 1 1 1 1 1
## [36] 2 2 1 2 1 1 2 1 2 1 1 1 2 1 2 1 1 2 1 2 1 1 2 1 2 1 2 1 1 1 2 1
## [71] 1 2 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 2 1 2 1 2 1 1 1 2 1 2 1 2 1 1
## [106] 2 1 2 1 1 1 2 1 1 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
## [141] 1 2 1 1 1 2 1 2 1 2 1 1 2 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 2 1 2 1 1
## [176] 1 1 2 1 1 2 1 1 1 2 1 1 2 1 2 1 2 1 1 1 1 1 2 1 2 1 1 2 1 2 1 1 2 1
## [211] 2 1 2 1 1 1 2 1 2 1 2 1 2 1 1 1 1 1 1 1 2 1 2 1 2 2 1 1 2 1 2 1 2 1
## [246] 1 2 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 1 2 2 1 1 2 1 2 1

# Visualisation de la densité et de la répartition des observations
plot(modu)
```

Histogram of Duration



Histogram of Waiting.Time



Données IRIS #

```
iris.mixmod = mixmodCluster(iris[-5], nbCluster = 1:9, model = mixmodGaussianModel())
```

```
summary(iris.mixmod)
```

```
## *****
```

```
## * Number of samples      = 150
```

```
## * Problem dimension      = 4
```

```
## *****
```

```
## *      Number of cluster = 3
```

```
## *      Model Type = Gaussian_p_Lk_Dk_A_Dk
```

```
## *      Criterion = BIC(553.4053)
```

```
## *      Parameters = list by cluster
```

```
## *      Cluster 1 :
```

```
##          Proportion = 0.3333
```

```
##          Means = 5.9177 2.7781 4.2106 1.3022
```

```
##          Variances = | 0.2274 0.0761 0.1487 0.0441 |
```

```
##                      | 0.0761 0.0807 0.0744 0.0347 |
```

```
##                      | 0.1487 0.0744 0.1692 0.0510 |
```

```
##                      | 0.0441 0.0347 0.0510 0.0342 |
```

```
## *      Cluster 2 :
```

```
##          Proportion = 0.3333
```

```
##          Means = 5.0060 3.4280 1.4620 0.2460
```

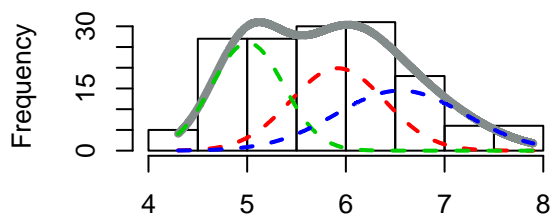
```
##          Variances = | 0.1328 0.1089 0.0192 0.0116 |
```

```
##                      | 0.1089 0.1545 0.0120 0.0100 |
```

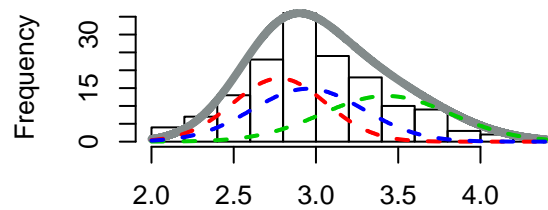
```
## | 0.0192 0.0120 0.0283 0.0058 |
## | 0.0116 0.0100 0.0058 0.0107 |
## * Cluster 3 :
## Proportion = 0.3333
## Means = 6.5516 2.9510 5.4909 1.9904
## Variances = | 0.4282 0.1078 0.3310 0.0630 |
## | 0.1078 0.1154 0.0878 0.0606 |
## | 0.3310 0.0878 0.3584 0.0831 |
## | 0.0630 0.0606 0.0831 0.0847 |
## * Log-likelihood = -186.5112
## *****
```

```
# Densité par classe
histCluster(iris.mixmod["bestResult"], iris[-5])
```

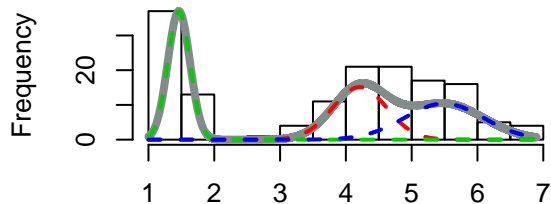
Histogram of Sepal.Length



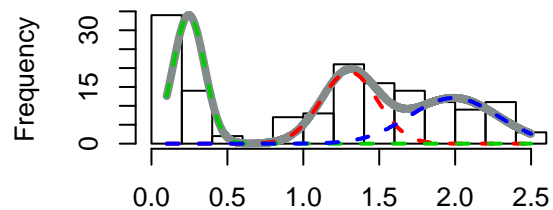
Histogram of Sepal.Width



Histogram of Petal.Length

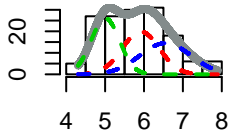


Histogram of Petal.Width

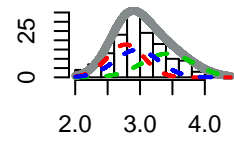
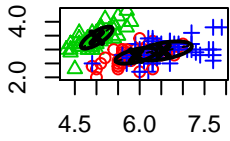


```
# Classification
plot(iris.mixmod)
```

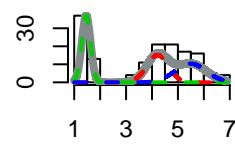
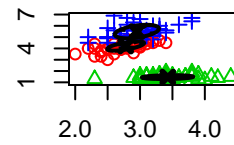
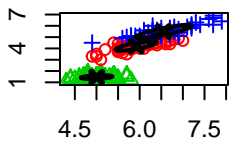
istogram of Sepal.Leng



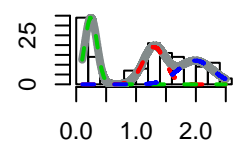
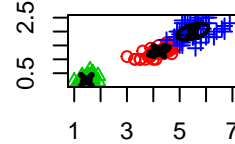
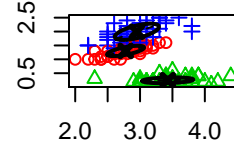
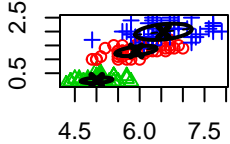
istogram of Sepal.Widt



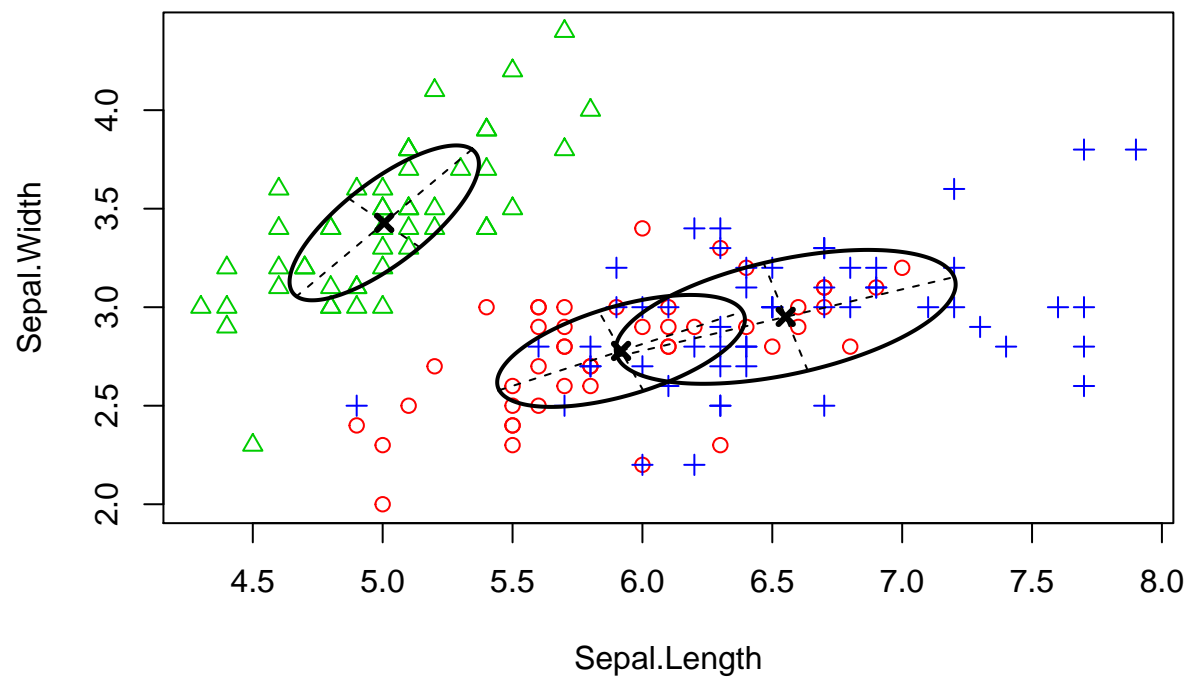
istogram of Petal.Leng



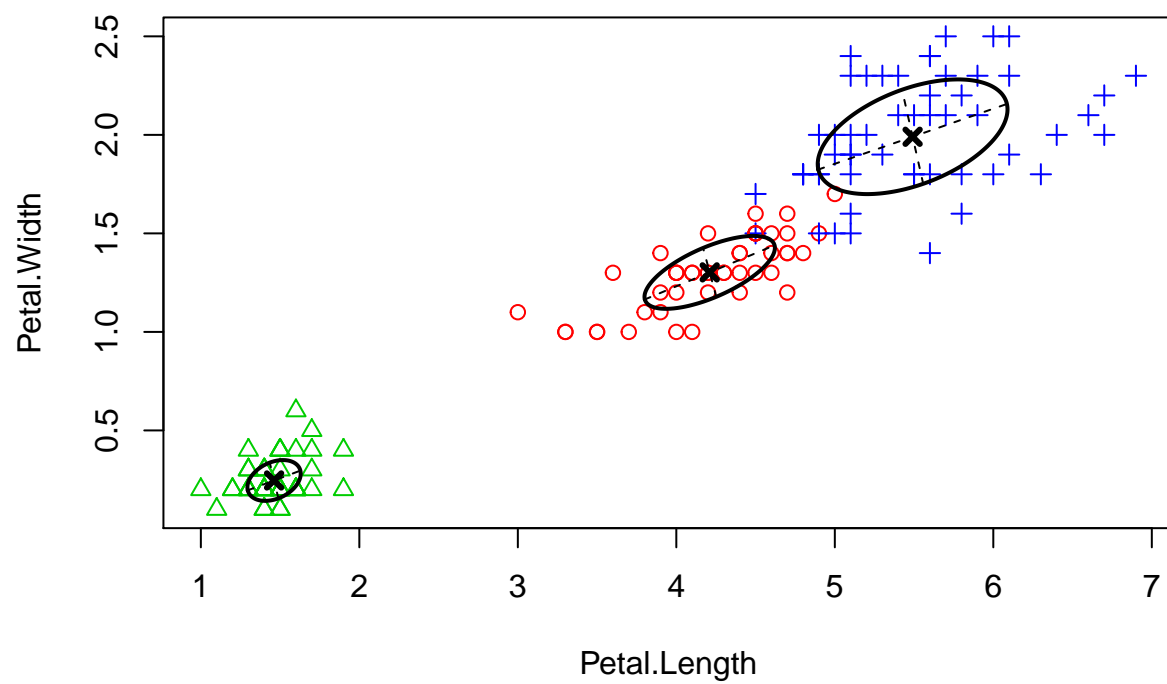
istogram of Petal.Widt



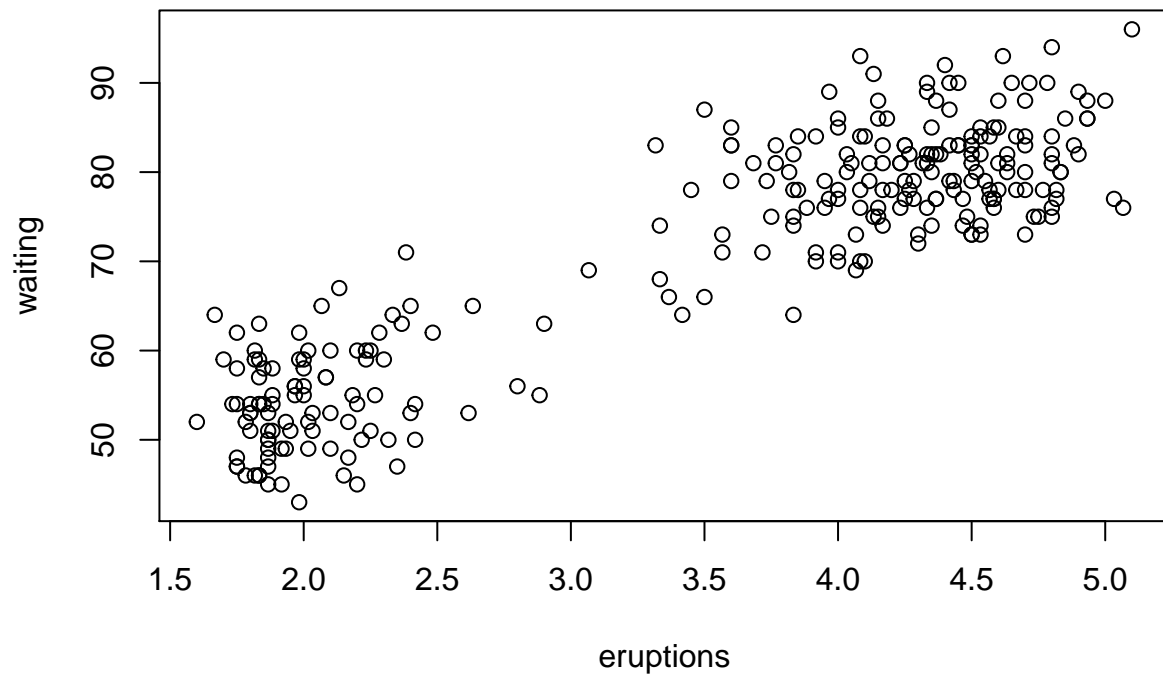
```
# Classification selon les variables Sepal.Length et Sepal.Width
plotCluster(iris.mixmod["bestResult"], iris[-5], variable1 = 1, variable2 = 2)
```



```
# Classification selon les variables Petal.Length et Petal.Width
plotCluster(iris.mixmod["bestResult"], iris[-5], variable1 = 3, variable2 = 4)
```



```
##### # Autre jeu de données avec utilisation mclust # #####  
plot(faithful)
```

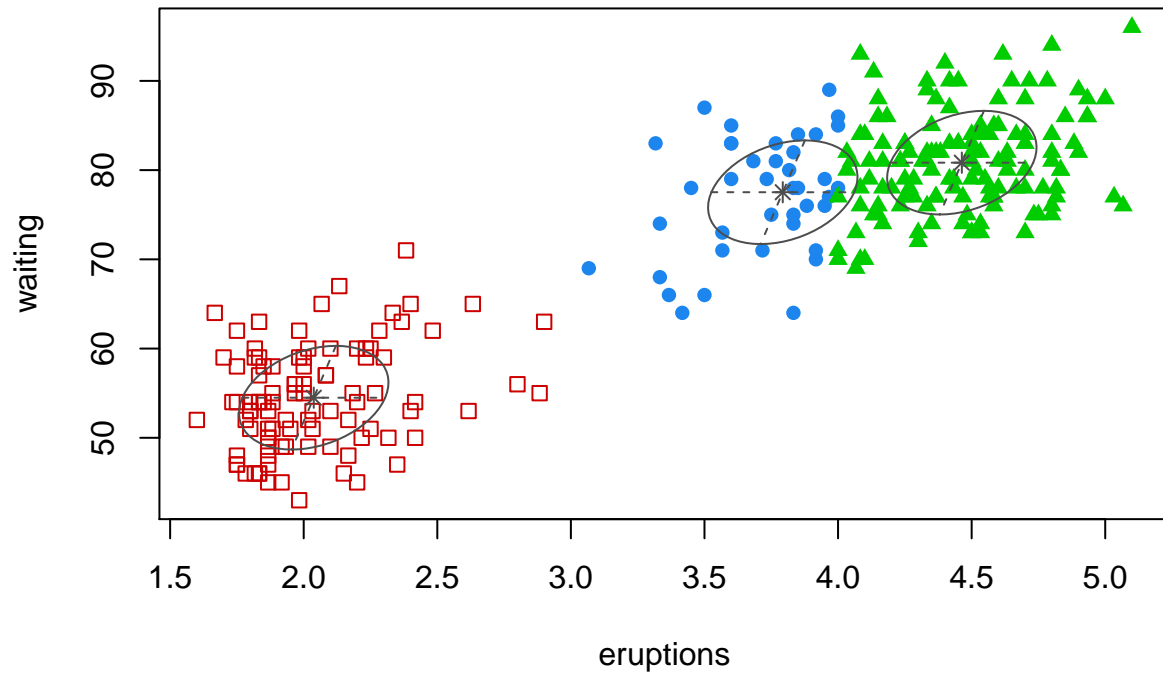
```
faithful.mclust = Mclust(faithful)
```

```
summary(faithful.mclust)
```

```
## -----
## Gaussian finite mixture model fitted by EM algorithm
## -----
##
## Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 3 components:
##
##   log.likelihood   n df       BIC       ICL
##         -1126.326 272 11 -2314.316 -2357.824
##
## Clustering table:
##    1  2  3
##   40 97 135
```

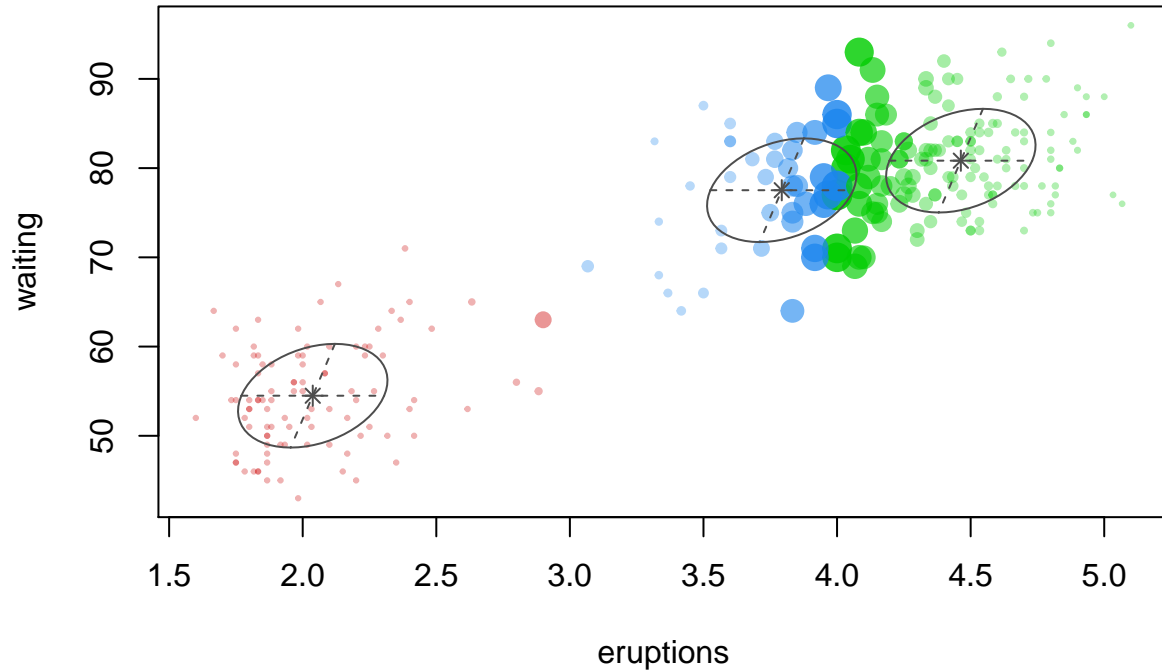
```
plot(faithful.mclust, what = "classification")
```

Classification



```
plot(faithful.mclust, what = "uncertainty")
```

Uncertainty



On un mélange composé de 5 gaussiennes différentes pour le graphe selon la première variable Duration.

On un mélange composé de 5 gaussiennes différentes pour le graphe selon la seconde variable Waiting.Time.

Il existe plusieurs modèles de mélanges, que se soient pour les mélanges gaussiens ou autres, dont certains ont beaucoup de paramètres libres donc sont plus flexibles mais complexes.

Comment choisir le bon modèle à utiliser ?

Critère BIC(Bayesian Information Criterion)

Le modèle m et le paramètre θ sont considérés comme des variables aléatoires.

On introduit $\Pi(\theta_m)$ loi a priori sur θ_m

$p(m)$ probabilité a priori d'un modèle m (\simeq poids que l'on veut attribuer à un modèle m)

On va chercher le modèle le plus vraisemblable, qui maximise :

$$p(m|x) = \frac{f(x|m)p(m)}{f(x)}$$

où x les observations

$p(m) = \frac{1}{|\mathcal{M}|}$ avec \mathcal{M} collection de modèles

$f(x)$ la densité des observations, elle ne dépend pas du modèle m

On veut donc maximiser la densité conditionnelle $f(x|m)$ aussi appelé **la vraisemblance intégrée** :

$$f(x|m) = \int f(x|\theta_m)\Pi(\theta_m)d\theta_m$$

sous certaines conditions : $\ln(f(x|m)) \simeq \ln(f(x|\hat{\theta}_m)) - \frac{v_n}{2} \ln(n)$ où $\hat{\theta}_m = \operatorname{argmax}_{[\theta_m]} f(x|\theta_m)$ et v_n nombre

de paramètres

on doit alors minimiser $BIC(m) = -2 \ln(f(x|\hat{\theta}_m)) \Pi(\theta_m) d\theta_m$

Pour résumé, on veut minimiser le critère d'information

$$BIC(m) = -2 \ln(L_{[m, \theta_m]}) + k \ln(N)$$

où $L_{[m, \theta_m]}$ est la vraisemblance du modèle à estimer, N le nombre d'observations dans l'échantillon et k le nombre de paramètres libres du modèle.

Le modèle M choisi sera tel que $BIC(M) = \operatorname{argmin}_{m \in \mathcal{M}} (BIC(m))$

L'avantage de ce critère est un bon comportement pour le choix du modèle.

Néanmoins, le nombre de classes peut être surestimées (on aura des classes imbriquées).

Plusieurs modèles et critère BIC

```
modu_BIC <- mixmodCluster(gey, nbCluster=2:5, criterion="BIC",
                          strategy=mixmodStrategy('EM'), models=mixmodGaussianModel())
modu_BIC
```

```
## *****
```

```
## *** INPUT:
```

```
## *****
```

```
## * nbCluster = 2 3 4 5
```

```
## * criterion = BIC
```

```
## *****
```

```
## *** MIXMOD Models:
```

```
## * list = Gaussian_pk_L_I Gaussian_pk_Lk_I Gaussian_pk_L_B Gaussian_pk_Lk_B Gaussian_pk_L_Bk Gaussian
```

```
## * This list includes models with free and equal proportions.
```

```
## *****
```

```
## * data (limited to a 10x10 matrix) =
```

```
##      Duration Waiting.Time
```

```
## [1,] 3.6      79
```

```
## [2,] 1.8      54
```

```
## [3,] 3.333    74
```

```
## [4,] 2.283    62
```

```
## [5,] 4.533    85
```

```
## [6,] 2.883    55
```

```
## [7,] 4.7      88
```

```
## [8,] 3.6      85
```

```
## [9,] 1.95     51
```

```
## [10,] 4.35    85
```

```
## * ... ..
```

```
## *****
```

```
## *** MIXMOD Strategy:
```

```
## * algorithm      = EM
```

```
## * number of tries = 1
```

```
## * number of iterations = 200
```

```
## * epsilon        = 0.001
```

```
## *** Initialization strategy:
```

```
## * algorithm      = smallEM
```

```
## * number of tries = 10
```

```
## * number of iterations = 5
```

```
## * epsilon        = 0.001
```

```
## * seed           = NULL
```

```
## *****
```

```
##
```

```

##
## *****
## *** BEST MODEL OUTPUT:
## *** According to the BIC criterion
## *****
## * nbCluster    = 3
## * model name   = Gaussian_p_L_C
## * criterion    = BIC(2312.6005)
## * likelihood   = -1131.0741
## *****
## *** Cluster 1
## * proportion = 0.3333
## * means      = 3.9763 78.7145
## * variances  = |    0.0798    0.5332 |
##               |    0.5332   34.2044 |
## *** Cluster 2
## * proportion = 0.3333
## * means      = 4.5545 81.0564
## * variances  = |    0.0798    0.5332 |
##               |    0.5332   34.2044 |
## *** Cluster 3
## * proportion = 0.3333
## * means      = 2.0390 54.5082
## * variances  = |    0.0798    0.5332 |
##               |    0.5332   34.2044 |
## *****

# Résumé des résultats
summary(modu_BIC)

## *****
## * Number of samples    = 272
## * Problem dimension    = 2
## *****
## *      Number of cluster = 3
## *      Model Type = Gaussian_p_L_C
## *      Criterion = BIC(2312.6005)
## *      Parameters = list by cluster
## *      Cluster 1 :
##           Proportion = 0.3333
##           Means = 3.9763 78.7145
##           Variances = |    0.0798    0.5332 |
##                       |    0.5332   34.2044 |
## *      Cluster 2 :
##           Proportion = 0.3333
##           Means = 4.5545 81.0564
##           Variances = |    0.0798    0.5332 |
##                       |    0.5332   34.2044 |
## *      Cluster 3 :
##           Proportion = 0.3333
##           Means = 2.0390 54.5082
##           Variances = |    0.0798    0.5332 |
##                       |    0.5332   34.2044 |
## *      Log-likelihood = -1131.0741
## *****

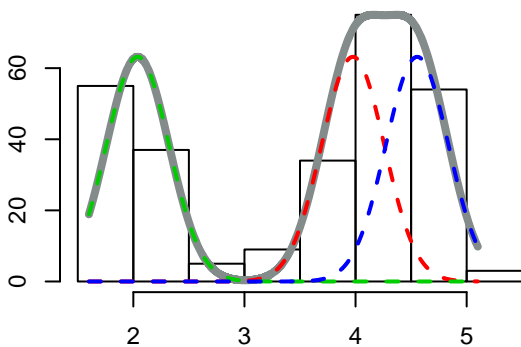
```

```
# Classification des individus
modu_BIC['partition']
```

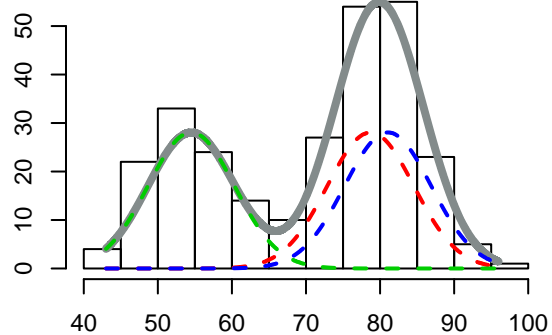
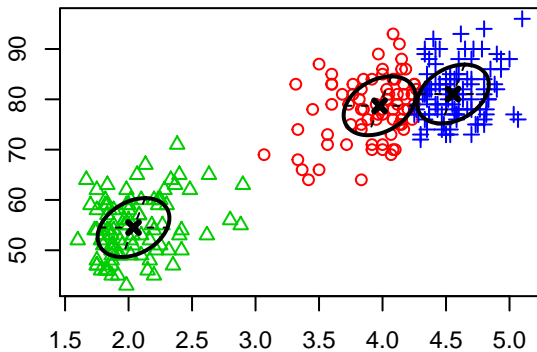
```
## [1] 1 3 1 3 2 3 2 1 3 2 3 1 1 3 2 3 3 2 3 1 3 3 1 1 2 1 3 1 1 2 2 2 1 1 1
## [36] 3 3 2 3 2 2 3 2 3 2 1 1 3 2 3 2 2 3 2 3 2 1 3 2 2 3 2 3 2 3 2 1 2 3 2
## [71] 1 3 2 1 3 2 3 2 1 1 1 2 1 3 1 2 1 2 3 1 3 2 3 2 3 2 2 1 3 2 3 2 3 2 1
## [106] 3 2 3 2 1 2 3 2 2 3 2 3 2 3 2 3 1 2 3 2 1 3 2 3 2 3 1 3 2 3 2 3 2 3 1
## [141] 1 3 2 2 2 3 2 3 2 3 2 1 3 2 1 1 2 1 3 1 3 1 3 1 1 2 3 2 3 2 3 3 2 1 1
## [176] 2 2 3 1 1 3 2 1 1 3 2 1 3 2 3 2 3 2 1 1 1 1 2 3 2 3 2 1 3 2 3 2 1 3 2
## [211] 3 2 3 1 1 1 3 2 3 1 3 1 3 2 1 1 1 2 1 2 1 3 1 3 2 3 3 2 1 3 1 3 2 3 2
## [246] 1 3 2 3 2 3 2 1 2 1 1 1 2 3 2 2 2 3 1 3 3 2 1 3 2 3 2
```

```
# Tri de la classification de mélange modu selon le critère BIC
tri_bic<-sortByCriterion(modu,"BIC")
plot(tri_bic)
```

Histogram of Duration



Histogram of Waiting.Time

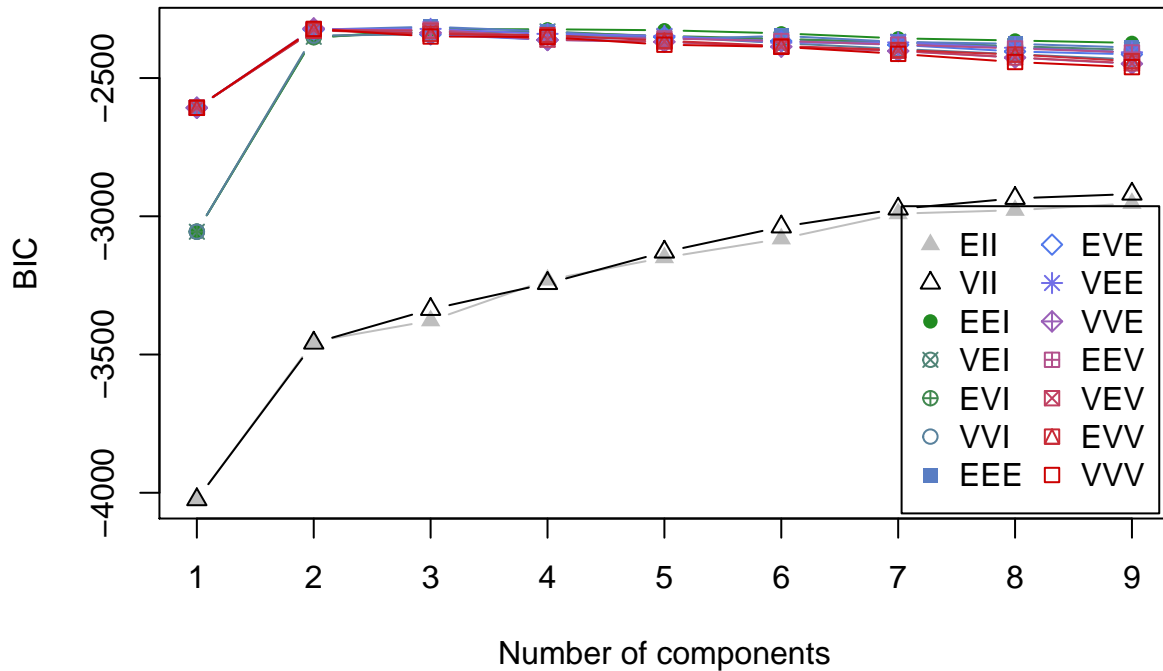


```
# Avec les données faithful
```

```
summary(faithful.mclust$BIC)
```

```
## Best BIC values:
##           EEE,3           VVE,2           VEE,3
## BIC      -2314.316 -2320.432980 -2322.103490
## BIC diff    0.000    -6.116684    -7.787194
```

```
plot(faithful.mclust, what = "BIC")
```



Critère ICL(Integrated Completed Likelihood)

Pour remédier au défaut du critère BIC qui peut engendrer des classes imbriquées du fait de la surrestimation, on utilise le critère ICL.

ICL est le critère BIC pénalisé par un terme d'entropie qui vise à prendre en compte l'imbrication des classes.

$$ICL(m) = BIC(m) - 2 \sum_{i=1}^n \sum_{k=1}^n \hat{z}_{ik} \ln(t_{ik}(\hat{\theta}_m))$$

où $entropie = 2 \sum_{i=1}^n \sum_{k=1}^n \hat{z}_{ik} \ln(t_{ik}(\hat{\theta}_m))$: pénalité dépendant des données, d'autant plus grande que les classes sont imbriquées.

et pour n assez grand \hat{z}_{ik} est défini par $\hat{z}_{ik} = 1$ si $k = \operatorname{argmax}_{[l]}(t_{il}(\hat{\theta}_m))$ et 0 sinon

```
# Plusieurs modèles et critère ICL
modu_ICL <- mixmodCluster(gey, nbCluster=2:5, criterion="BIC",
                          strategy=mixmodStrategy('EM'), models=mixmodGaussianModel())
modu_ICL
```

```
## *****
## *** INPUT:
## *****
## * nbCluster = 2 3 4 5
## * criterion = BIC
## *****
```

```

## *** MIXMOD Models:
## * list = Gaussian_pk_L_I Gaussian_pk_Lk_I Gaussian_pk_L_B Gaussian_pk_Lk_B Gaussian_pk_L_Bk Gaussian
## * This list includes models with free and equal proportions.
## *****
## * data (limited to a 10x10 matrix) =
##      Duration Waiting.Time
## [1,] 3.6      79
## [2,] 1.8      54
## [3,] 3.333    74
## [4,] 2.283    62
## [5,] 4.533    85
## [6,] 2.883    55
## [7,] 4.7      88
## [8,] 3.6      85
## [9,] 1.95     51
## [10,] 4.35    85
## * ... ...
## *****
## *** MIXMOD Strategy:
## * algorithm      = EM
## * number of tries = 1
## * number of iterations = 200
## * epsilon        = 0.001
## *** Initialization strategy:
## * algorithm      = smallEM
## * number of tries = 10
## * number of iterations = 5
## * epsilon        = 0.001
## * seed           = NULL
## *****
##
##
## *****
## *** BEST MODEL OUTPUT:
## *** According to the BIC criterion
## *****
## * nbCluster      = 3
## * model name     = Gaussian_p_L_C
## * criterion      = BIC(2312.6005)
## * likelihood     = -1131.0741
## *****
## *** Cluster 1
## * proportion = 0.3333
## * means      = 3.9763 78.7250
## * variances  = |    0.0798    0.5350 |
##               |    0.5350    34.2187 |
## *** Cluster 2
## * proportion = 0.3333
## * means      = 2.0390 54.5083
## * variances  = |    0.0798    0.5350 |
##               |    0.5350    34.2187 |
## *** Cluster 3
## * proportion = 0.3333
## * means      = 4.5545 81.0479

```



```
## * variances = |      0.0798      0.5350 |
##               |      0.5350     34.2187 |
## *****

# Résumé des résultats
summary(modu_ICL)

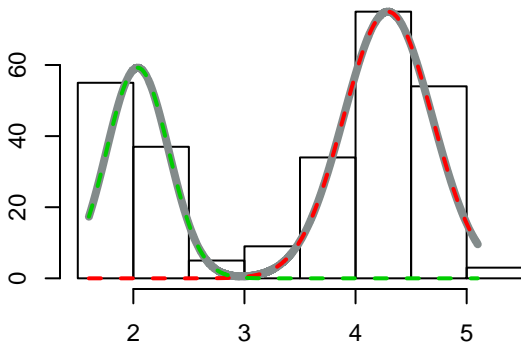
## *****
## * Number of samples      = 272
## * Problem dimension      = 2
## *****
## *      Number of cluster = 3
## *      Model Type = Gaussian_p_L_C
## *      Criterion = BIC(2312.6005)
## *      Parameters = list by cluster
## *      Cluster 1 :
##               Proportion = 0.3333
##               Means = 3.9763 78.7250
##               Variances = |      0.0798      0.5350 |
##                           |      0.5350     34.2187 |
## *      Cluster 2 :
##               Proportion = 0.3333
##               Means = 2.0390 54.5083
##               Variances = |      0.0798      0.5350 |
##                           |      0.5350     34.2187 |
## *      Cluster 3 :
##               Proportion = 0.3333
##               Means = 4.5545 81.0479
##               Variances = |      0.0798      0.5350 |
##                           |      0.5350     34.2187 |
## *      Log-likelihood = -1131.0741
## *****

# Classification des individus
modu_ICL['partition']

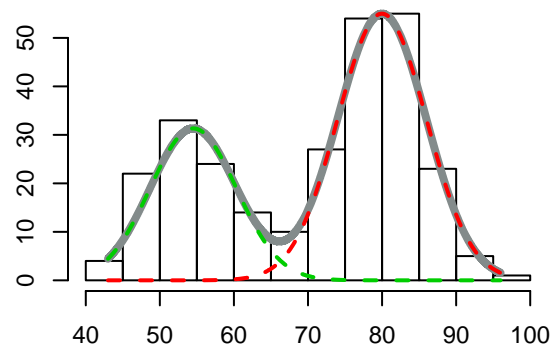
## [1] 1 2 1 2 3 2 3 1 2 3 2 1 1 2 3 2 2 3 2 1 2 2 1 1 3 1 2 1 1 3 3 3 1 1 1
## [36] 2 2 3 2 3 3 2 3 2 3 1 1 2 3 2 3 3 2 3 2 3 1 2 3 3 2 3 2 3 2 3 1 3 2 3
## [71] 1 2 3 1 2 3 2 3 1 1 1 3 1 2 1 3 1 3 2 1 2 1 2 3 2 3 3 1 2 3 2 3 2 3 1
## [106] 2 3 2 3 1 3 2 3 3 2 3 2 3 2 3 2 1 3 2 3 1 2 3 2 3 2 1 2 3 2 3 2 3 2 1
## [141] 1 2 3 3 3 2 3 2 3 2 3 1 2 3 1 1 3 1 2 1 2 1 2 1 1 3 2 3 2 3 2 2 3 1 1
## [176] 3 3 2 1 1 2 3 1 1 2 3 1 2 3 2 3 2 3 1 1 1 1 3 2 3 2 3 1 2 3 2 3 1 2 3
## [211] 2 3 2 1 1 1 2 3 2 1 2 1 2 3 1 1 1 3 1 3 1 2 1 2 3 2 2 3 1 2 1 2 3 2 3
## [246] 1 2 3 2 3 2 3 1 3 1 1 1 3 2 3 3 3 2 1 2 2 3 1 2 3 2 3

# Tri de la classification de mélange modu selon le critère BIC
tri_icl<-sortByCriterion(modu,"ICL")
plot(tri_icl)
```

Histogram of Duration



Histogram of Waiting.Time

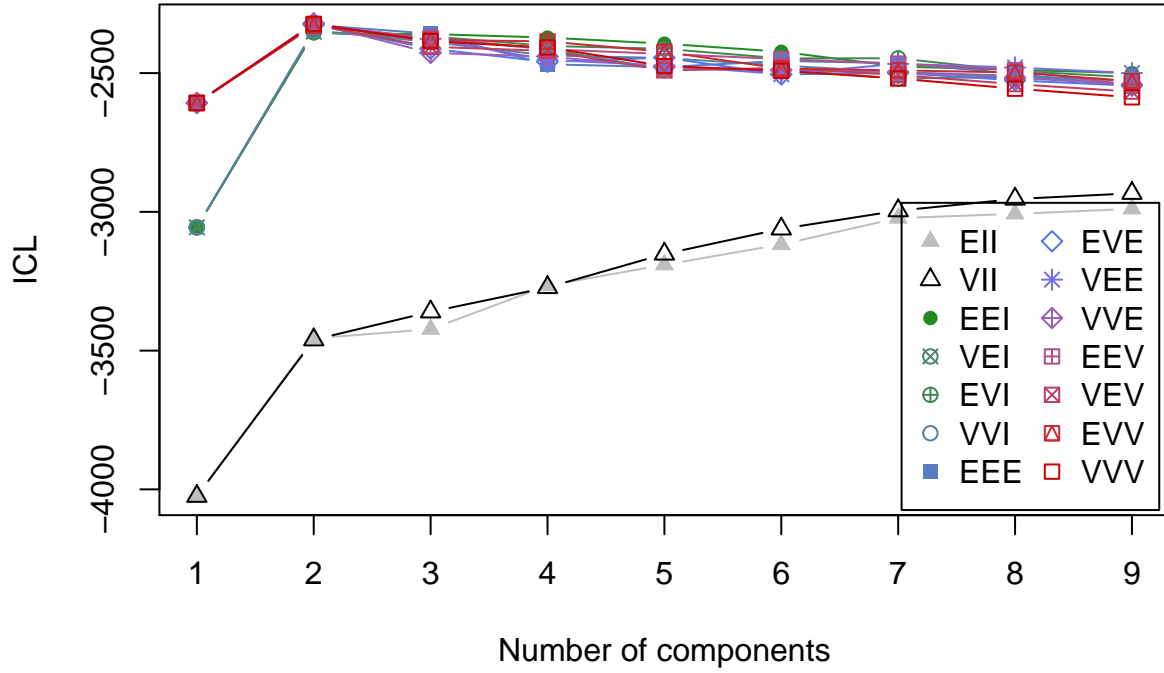


Avec les données faithful

```
faithful.mclustICL = mclustICL(faithful)
summary(faithful.mclustICL)
```

```
## Best ICL values:
##           VVE,2           VVV,2           VEE,2
## ICL      -2320.763 -2322.697467 -2323.39551
## ICL diff      0.000   -1.934645   -2.63269
```

```
plot(faithful.mclustICL)
```



Algorithme CEM : Classification EM

Ajout d'une étape de classification dans **EM**

- 1) Déterminer une situation initiale
- 2) **Estimation** des probabilités a posteriori t_{ik} (identique)
- 3) **Classification** des individus avec la méthode du MAP

$$z_k = \{i | t_{ik} = \max_{l=1, \dots, s} t_{il}\}$$

- 4) **Maximisation** : calcul des paramètres du mélange

$$p_k = \frac{\text{Card}(z_k)}{n}$$

a_k = dépendant du modèle

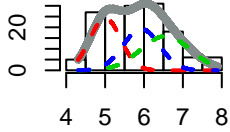
- 5) Itérer les étapes 2 à 4, jusqu'à la convergence (évolution très faible de L_c)

Comparaison entre EM et CEM

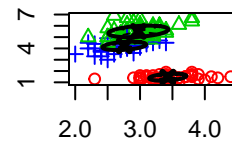
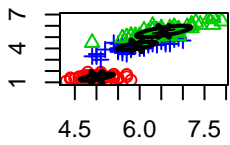
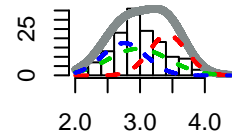
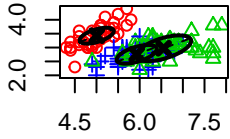
```
iris.EM = mixmodCluster(iris[-5], 3, strategy = mixmodStrategy("EM", 20, "random"))
```

```
plot(iris.EM)
```

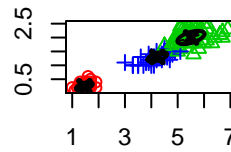
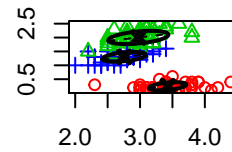
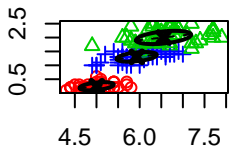
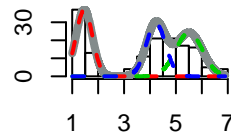
istogram of Sepal.Leng



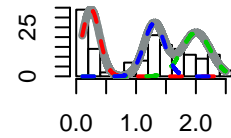
istogram of Sepal.Widt



istogram of Petal.Leng

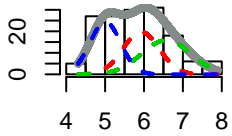


istogram of Petal.Widt

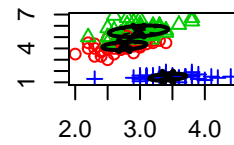
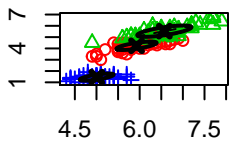
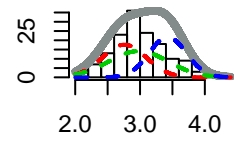
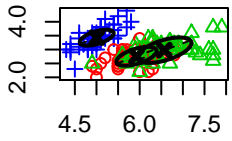


```
iris.CEM = mixmodCluster(iris[-5], 3, strategy = mixmodStrategy("CEM", 20, "random"))
plot(iris.CEM)
```

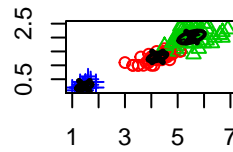
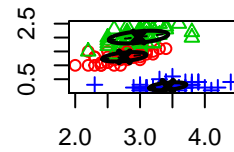
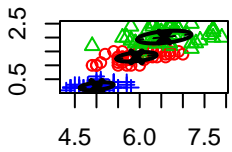
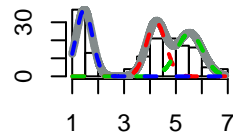
istogram of Sepal.Leng



istogram of Sepal.Widt



istogram of Petal.Leng



istogram of Petal.Widt

