TP n?1 - Mod?les lin?aires et mod?les de m?langes

Salim Nadir et Guillaume Ostrom

Application I : mod?le de r?gression lin?raire

Analyse des donn?es du probleme

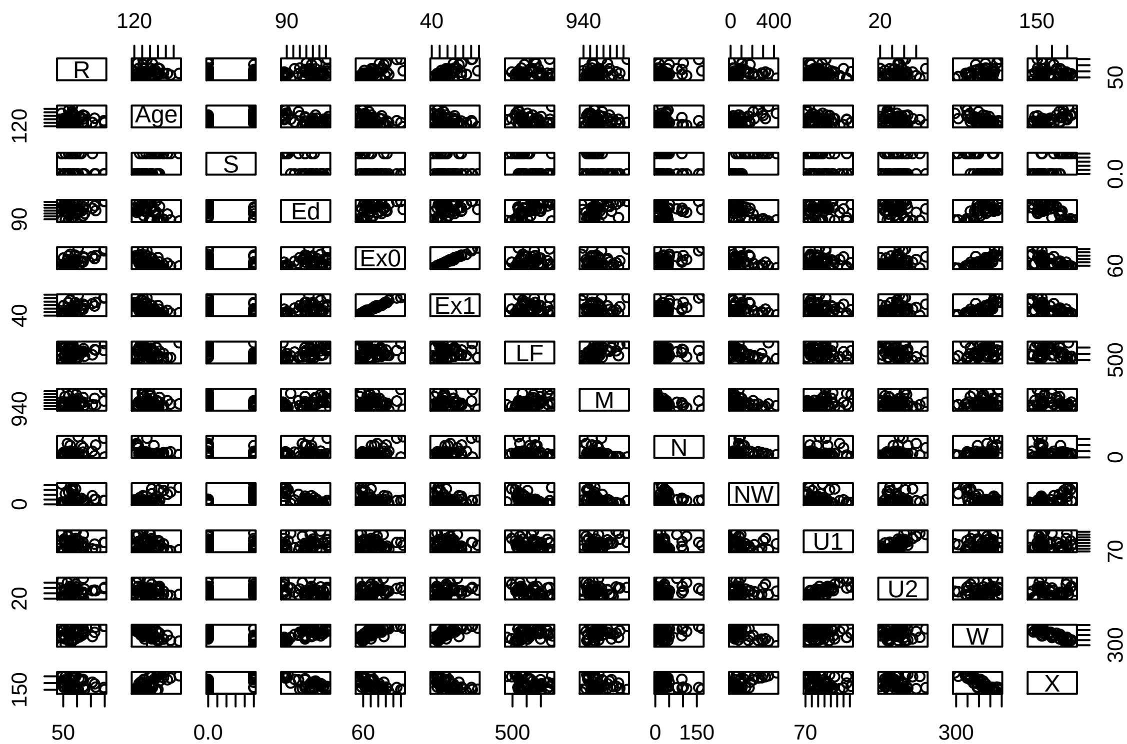
Nous constatons que le fichier *USCrimeinfo.txt* d?crit les diff?rentes variables pr?sentes dans le fichier *UsCrime.txt*, o? la premi?re colonne (R) est la variable cible.

Proc?dons au chargement du fichier *UsCrime.txt* :

```
tab = read.table('UsCrime.txt',header=TRUE)
```

Puis visualisons les nuages de points entre les variables :

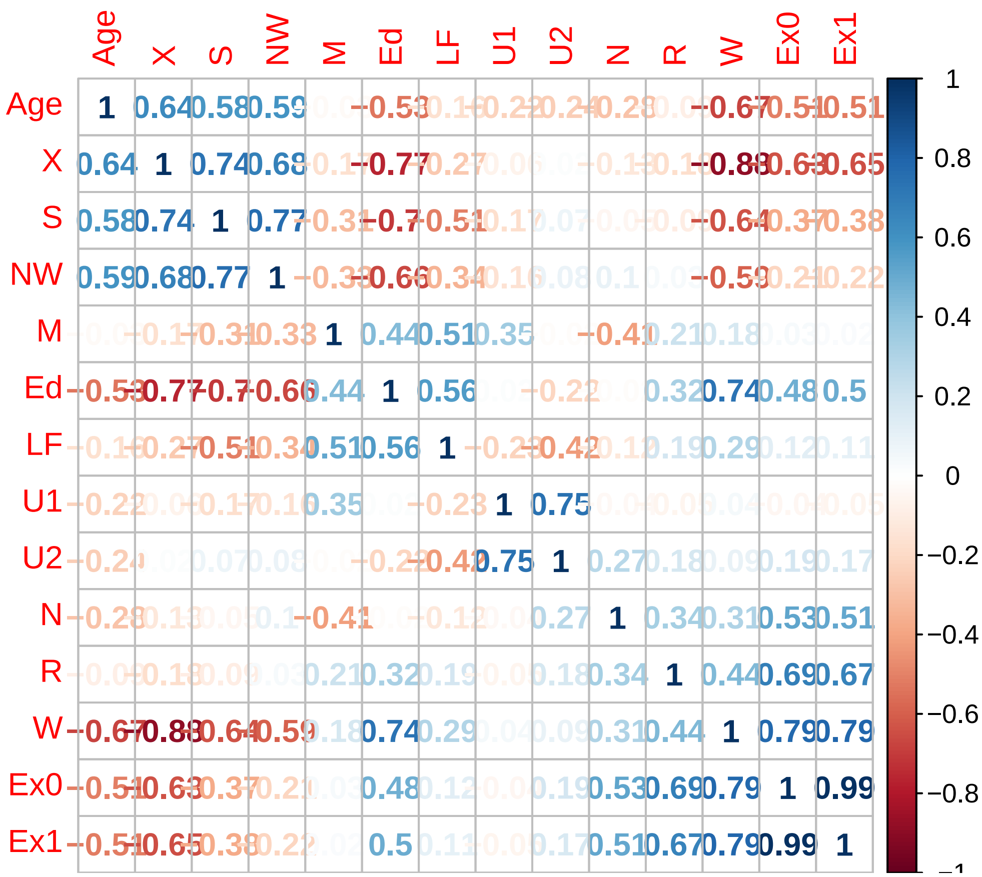
```
pairs(tab)
```



Parmis les 182 observations disponibles, nous constatons une forte corr?lation positive entre les variables **Ex0** et **Ex1** ainsi qu'entre les variables **U1** et **U2**. Cependant les variables **W** et **X** ont aussi une forte corr?lation mais celle-ci est n?gative. D'autres sont moins marqu?es tel que **W** et **Ex1** par exemple.

Int?ressons nous maintenant aux corr?lations entre les diff?rentes variables et v?rifions les intuitions pr?c?dentes.

```r
library(corrplot)
corrplot(cor(tab),order="hclust", method="number")
```



Les co?fficients de corr?lation confirment bien ce que nous avons observ? avec les nuages de points. Nous pouvons relever d'autres correlation int?ressantes, tel que **X** et **Ed** qui ont une corr?lation n?gative.

Mod?le lin?aire

Nous souhaitons ?tudier un mod?le lin?aire. Formellement, celui-ci nous permettra d'expliciter la variable cible en fonction des autres variables disponibles.

1. Ex?cution du mod?le :

```r
res=lm('R~.',data=tab)
print(res)
```

```
##
## Call:
## lm(formula = "R~.", data = tab)
##
```

```
## Coefficients:
## (Intercept)          Age            S           Ed          Ex0
##  -6.918e+02    1.040e+00   -8.308e+00    1.802e+00    1.608e+00
##         Ex1           LF            M            N           NW
##  -6.673e-01   -4.103e-02    1.648e-01   -4.128e-02    7.175e-03
##          U1           U2            W            X
##  -6.017e-01    1.792e+00    1.374e-01    7.929e-01
```

```r
summary(res)
```

```
##
## Call:
## lm(formula = "R~.", data = tab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.884 -11.923  -1.135  13.495  50.560
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.918e+02  1.559e+02  -4.438 9.56e-05 ***
## Age          1.040e+00  4.227e-01   2.460  0.01931 *
## S           -8.308e+00  1.491e+01  -0.557  0.58117
## Ed           1.802e+00  6.496e-01   2.773  0.00906 **
## Ex0          1.608e+00  1.059e+00   1.519  0.13836
## Ex1         -6.673e-01  1.149e+00  -0.581  0.56529
## LF          -4.103e-02  1.535e-01  -0.267  0.79087
## M            1.648e-01  2.099e-01   0.785  0.43806
## N           -4.128e-02  1.295e-01  -0.319  0.75196
## NW           7.175e-03  6.387e-02   0.112  0.91124
## U1          -6.017e-01  4.372e-01  -1.376  0.17798
## U2           1.792e+00  8.561e-01   2.093  0.04407 *
## W            1.374e-01  1.058e-01   1.298  0.20332
## X            7.929e-01  2.351e-01   3.373  0.00191 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.94 on 33 degrees of freedom
## Multiple R-squared:  0.7692, Adjusted R-squared:  0.6783
## F-statistic: 8.462 on 13 and 33 DF,  p-value: 3.686e-07
```

```r
attributes(res)
```

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

Nous obtenons un mod?le ? 33 degr?s de libert?. Nous constatons que la $p-value$ de la statistique de Fisher est tr?s faible par rapport ? $alpha$ ce qui rejette l'hypothese **H0 des co?fficients directeurs nuls** et qui confirme bien l'existance de **H1 coefficients non nuls** pour la r?gression lin?aire ?tudi?e.

Ajoutons que l'offset **(Intercept)** a une $p-value$ tr?s faible (10^{-5}) ce qui implique le rejet de **H0** ce qui confirme son co?fficient directeur tr?s fort. Aussi, nous remarquons que les variable **Ed** poss?de l'un des co?fficient directeur les plus ?lev? et a une $p-value$ faible. Ce qui permet de rejetter l'hypoth?se **H0** sur Ed. Enfin **X** a une $p-value$ faible. Ce qui implique le rejet de l'hypoth?se **H0**

2. Le mod?le global :

R^2 est le coefficient de d?termination mesurant la qualit? de la pr?diction d'une r?gression lin?aire. Plus R^2 est proche de 1 plus les donn?es sont proches de la droite de r?gression. Soit y_i les valeurs des donn?es, $\hat{y}_i$ les valeurs pr?dites et $\bar{y}$ la moyenne des mesures, alors:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

R^2 exprime la colin?arit? de y_i et $\hat{y}_i$. Il s'agit en fait du $cos^2(y_i, \hat{y}_i)$. Dans notre mod?le $R^2 = 0.7692$ Notons que R^2 est ?lev? mais celui-ci n'est pas suffisement pertinant car le nombre de dimensions (**14**) est comparable au nombre d'entr?es (**48**). Ainsi $R^2 ajust?$ est plus adatp? ? notre mod?le.

D?terminer si le mod?le est significatif revient ? ?valuer si la F-statistique du test de Fisher est pertinante : Soit β_i les coefficients de notre mod?le.

$$H_0 : \beta_1 = ... = \beta_n = 0$$

$$H_1 : \exists \text{ i tel que } \beta_i \neq 0$$

La F-statistique obtenue est 8.462 associ?e ? la $p-value = 3.686(10^{-7})$. La $p-value$?tant faible devant 0.05 on peut donc dire que le mod?le est significatif.

3. Les coefficients du mod?le

Dans notre ?tude, le test statistique utilis? est un test de Student $t-value$. La t-statistique est le rapport entre la valeur du coefficient *Estimate* et son ?cart type *Std.Error*. Enfin la probabilit? d'atteindre une t-value sup?rieure est d?crite par la $p-value$. Plus la $p-value$ est faible plus l'on peut rejetter H_0 et ainsi consid?rer le coefficient comme significatif. Nous en d?duisons que les ast?risques indiquent le niveau de confiance avec lequel nous pouvons rejetter H_0, ? 99.9% pour $***$, ? 99% pour $**$ et 95% pour $*$.

L'?tude des $p-values$ de chacune des variables nous permet de mettre en ?vidence la significativit? de chacune de ces variables dans notre mod?le global.

On peut en conclure que 4 variables sont significatives dans notre mod?le: **Ed**, **X**, **Age** et **U2**.

Voici les intervalles de confiance pour chacun des coefficients au risque 5% :

```
confint(res,level = 0.95)
```

```
##                     2.5 %       97.5 %
## (Intercept) -1.008994e+03 -374.6812339
## Age          1.798032e-01    1.8998161
## S           -3.864617e+01   22.0295401
## Ed           4.798774e-01    3.1233247
## Ex0         -5.460558e-01    3.7616925
## Ex1         -3.004455e+00    1.6699389
## LF          -3.532821e-01    0.2712200
## M           -2.623148e-01    0.5919047
## N           -3.047793e-01    0.2222255
```

```
## NW          -1.227641e-01    0.1371135
## U1          -1.491073e+00    0.2877222
## U2           5.049109e-02    3.5340347
## W           -7.795484e-02    0.3526718
## X            3.146483e-01    1.2712173
```

on observe que les intervalles de confiances de nos variables sont les plus resserr?s.

N.B.: au risque 1% :

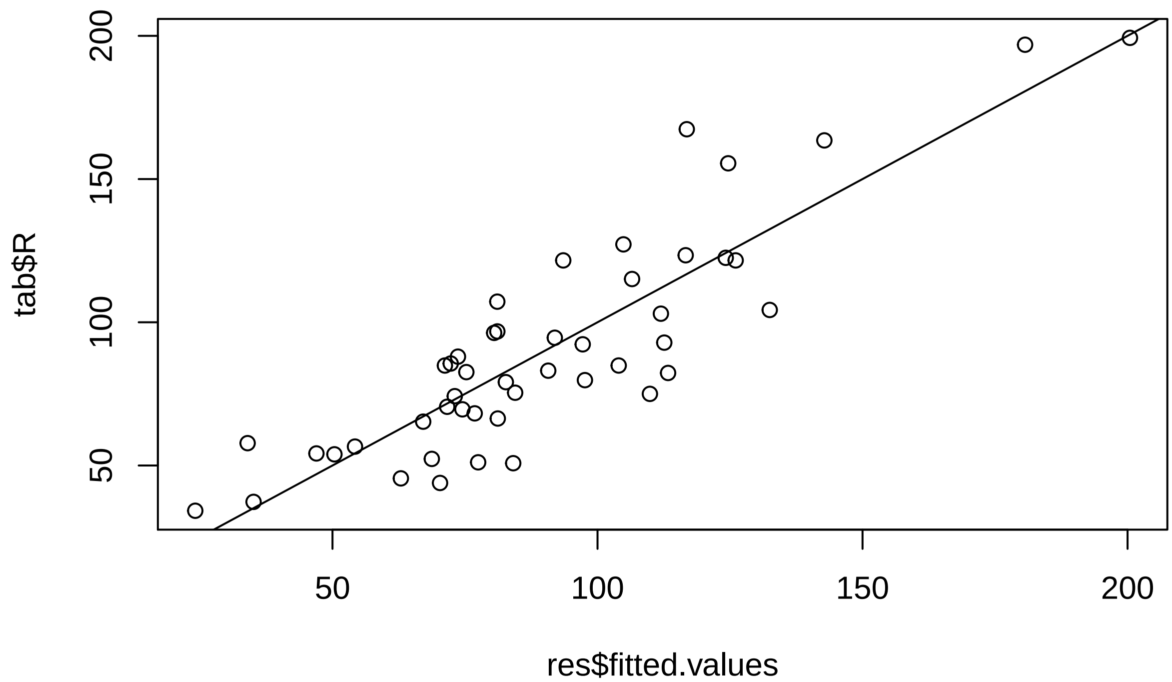
```r
confint(res,level = 0.99)
```

```
##                     0.5 %        99.5 %
## (Intercept) -1117.9223933 -265.7527826
## Age            -0.1155691    2.1951884
## S             -49.0658069   32.4491811
## Ed              0.0259268    3.5772753
## Ex0            -1.2858113    4.5014481
## Ex1            -3.8071739    2.4726574
## LF             -0.4605258    0.3784637
## M              -0.4090071    0.7385970
## N              -0.3952801    0.3127264
## NW             -0.1673920    0.1817414
## U1             -1.7965395    0.5931889
## U2             -0.5477265    4.1322523
## W              -0.1519049    0.4266219
## X               0.1503798    1.4354858
```

4. Etude des valeurs predites

Affichage des pr?dictions de la variable cible en fonction des valeurs observ?es :

```r
plot(res$fitted.values,tab$R)
lines(0:250, 0:250)
```



Nous constatons que le r?sultat est relativement satisfaisant. Les donn?es ne sont pas trop ?loign?es de la pr?diction.

Voici les intervalles de confiance pour les valeurs pr?dites au risque 5% :

```
predict(res,tab, interval = 'predict')
```

```
##          fit        lwr       upr
## 1   82.70143  33.807919 131.59493
## 2  142.79676  94.406373 191.18715
## 3   33.98746 -15.807593  83.78252
## 4  180.66596 127.563662 233.76826
## 5  116.63068  64.483571 168.77778
## 6   76.82991  26.518173 127.14165
## 7   80.49951  27.565722 133.43330
## 8  124.65377  72.110760 177.19677
## 9   72.30419  24.466227 120.14216
## 10  71.63495  23.044864 120.22504
## 11 116.83976  66.696540 166.98298
## 12  71.20484  23.758018 118.65167
## 13  77.48311  28.067020 126.89919
## 14  81.18429  31.966865 130.40172
## 15  97.62269  44.959887 150.28548
## 16  91.94335  42.202607 141.68409
## 17  50.36998   1.231739  99.50821
## 18 112.59273  61.514787 163.67068
```

```
## 19 109.88397  57.969606 161.79833
## 20 124.19286  73.783773 174.60195
## 21  73.06820  25.004062 121.13233
## 22  70.28960  16.659006 123.92019
## 23  93.53601  42.473928 144.59809
## 24  81.10478  31.465405 130.74415
## 25  68.73953  18.784339 118.69473
## 26 200.45494 144.454772 256.45512
## 27  24.11037 -25.876879  74.09762
## 28 126.06171  74.900672 177.22275
## 29 132.49024  78.524319 186.45617
## 30  74.51572  24.403279 124.62815
## 31  35.11199 -19.343630  89.56760
## 32  84.46111  34.334667 134.58755
## 33  81.09615  31.278635 130.91366
## 34  97.20686  49.110209 145.30351
## 35  67.11134  15.775461 118.44722
## 36 104.89276  50.953854 158.83167
## 37  90.70442  32.941164 148.46767
## 38  54.24417   4.620145 103.86820
## 39  75.25593  27.550081 122.96177
## 40 106.51724  56.015065 157.01942
## 41  73.67235  23.379572 123.96512
## 42  46.96884  -2.209203  96.14688
## 43 113.31446  61.738490 164.89042
## 44 111.95800  63.060025 160.85597
## 45  62.90137   9.205436 116.59730
## 46  84.09636  34.687618 133.50509
## 47 103.99337  52.359525 155.62722
```

On constate que la pr?diction est l?g?rement corr?l?e ? la variable **R** ? pr?dire. Cependant les biais mettent en ?vidence un ?talement.

```r
predict(res,tab, interval = 'confidence', level = 0.95)
```

```
##          fit        lwr       upr
## 1   82.70143  62.729456 102.67340
## 2  142.79676 124.090238 161.50328
## 3   33.98746  11.900088  56.07484
## 4  180.66596 151.888215 209.44371
## 5  116.63068  89.656149 143.60520
## 6   76.82991  53.601147 100.05867
## 7   80.49951  52.033907 108.96512
## 8  124.65377  96.921618 152.38592
## 9   72.30419  55.077020  89.53137
## 10  71.63495  52.417741  90.85217
## 11 116.83976  93.978287 139.70123
## 12  71.20484  55.095662  87.31402
## 13  77.48311  56.263916  98.70229
## 14  81.18429  60.431950 101.93664
## 15  97.62269  69.664237 125.58113
## 16  91.94335  69.978699 113.90800
## 17  50.36998  29.806151  70.93380
```

```
## 18 112.59273  87.747994 137.43747
## 19 109.88397  83.362178 136.40575
## 20 124.19286 100.753993 147.63173
## 21  73.06820  55.222596  90.91380
## 22  70.28960  40.548300 100.03089
## 23  93.53601  68.723901 118.34812
## 24  81.10478  59.370660 102.83890
## 25  68.73953  46.293467  91.18560
## 26 200.45494 166.627089 234.28280
## 27  24.11037   1.593050  46.62770
## 28 126.06171 101.046588 151.07683
## 29 132.49024 102.148433 162.83205
## 30  74.51572  51.721839  97.30959
## 31  35.11199   3.907523  66.31645
## 32  84.46111  61.636457 107.28576
## 33  81.09615  58.958195 103.23410
## 34  97.20686  79.273862 115.13986
## 35  67.11134  41.740549  92.48213
## 36 104.89276  74.599027 135.18650
## 37  90.70442  54.031568 127.37726
## 38  54.24417  32.545124  75.94323
## 39  75.25593  58.399111  92.11274
## 40 106.51724  82.878833 130.15565
## 41  73.67235  50.484686  96.86000
## 42  46.96884  26.310079  67.62760
## 43 113.31446  87.461318 139.16759
## 44 111.95800  91.975095 131.94090
## 45  62.90137  33.042406  92.76033
## 46  84.09636  62.894283 105.29843
## 47 103.99337  78.024955 129.96179
```

5. Etude des r?sidus

Voici l'erreur quadratique des r?sidus :

```r
mean((res$residuals-mean(res$residuals))^2)
```

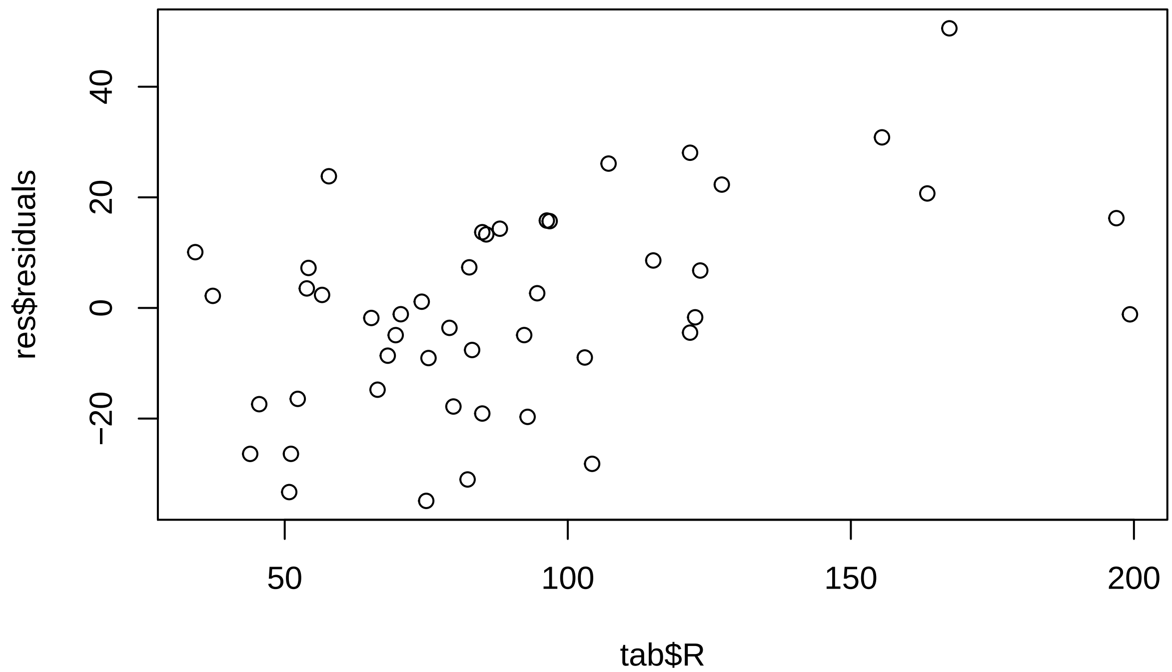
```
## [1] 337.8447
```

Voici l'?stimation r?siduelle de la variance quadratique :

```r
var(res$residuals)
```

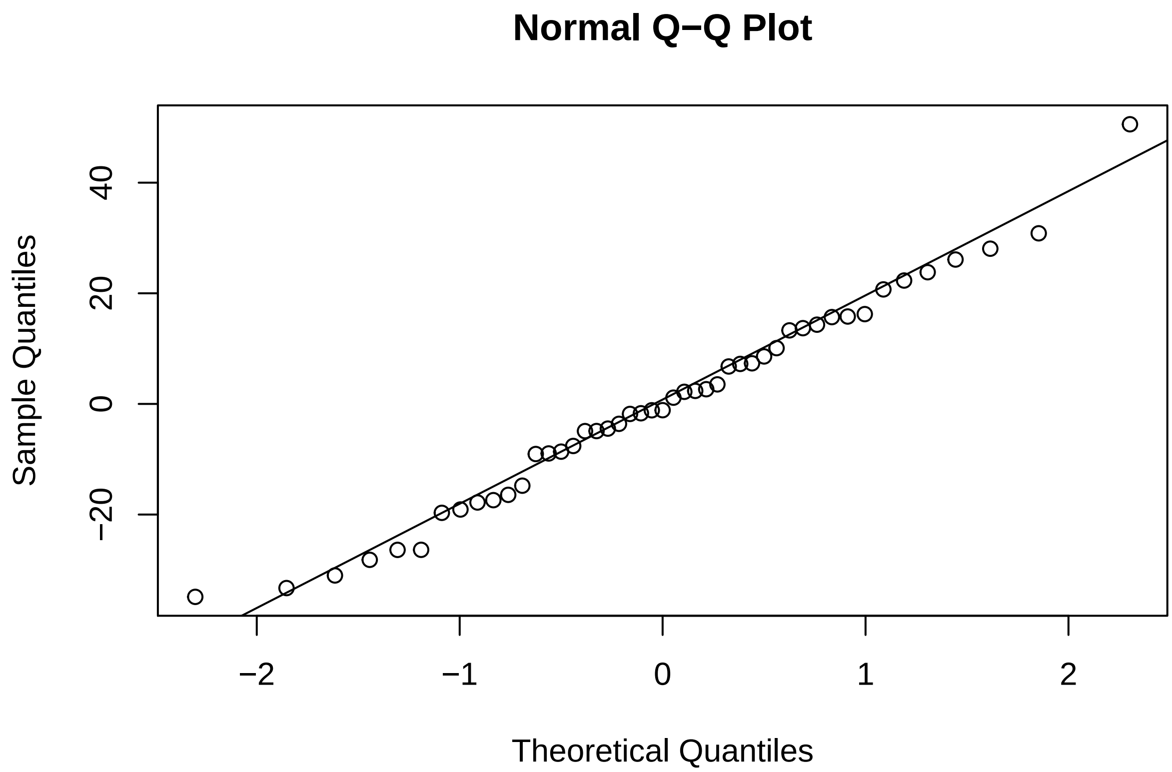
```
## [1] 345.1891
```

```r
plot(tab$R,res$residuals)
```



Nous observons que les valeurs des r?sidus sont proches de la valeur 0. On peut donc en conclure que le mod?le est relativement satisfaisant, cependant il est possible de l'affiner.

```r
qqnorm(res$residuals)

qqline(res$residuals)
```



Nous observons que les r?sidus semblent suivrent une loi normale. Nous utilisons le test de Shapiro sur les r?sidus afin de confirmer cette observation.

```
shapiro.test(res$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res$residuals
## W = 0.98553, p-value = 0.8216
```

Nous constatons que la $p - value$ est tr?s grande donc on ne rejette pas l'hypoth?se de normalit?.

6. Performances du mod?le sur de nouvelles donnees

N.B.: Correction dans l'?nonc? : indTest croit de 3 en 3 depuis 1 : 1,4,7...

Nous souhaitons entrainer notre mod?le sur deux tiers des donn?es et tester notre mod?le sur le tiers des donn?es restante.

Nous g?n?rons les indices de test :

```
indTest<-seq(1,nrow(tab),by = 3)
indTrain<-setdiff(1:nrow(tab),indTest)
```

```
tabTest<-tab[indTest,]
tabTrain<-tab[indTrain,]
```

Import des donn?es:

```
res<-lm('R~.',data=tabTrain)
```

```
summary(res)
```

```
##
## Call:
## lm(formula = "R~.", data = tabTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.121  -8.105  -1.987   5.332  47.276
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -369.94395  215.46161  -1.717  0.10414
## Age            0.93812    0.61065   1.536  0.14287
## S            -27.67250   19.87275  -1.392  0.18172
## Ed             2.05416    1.02803   1.998  0.06195 .
## Ex0            1.69651    1.24210   1.366  0.18979
## Ex1           -0.54224    1.45121  -0.374  0.71328
## LF             0.16724    0.32224   0.519  0.61045
## M             -0.26281    0.32182  -0.817  0.42544
## N             -0.33196    0.18566  -1.788  0.09161 .
## NW             0.09722    0.10607   0.917  0.37216
## U1            -0.74737    0.51157  -1.461  0.16227
## U2             2.20517    0.90823   2.428  0.02658 *
## W              0.04397    0.12754   0.345  0.73451
## X              0.86619    0.29557   2.931  0.00934 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.57 on 17 degrees of freedom
## Multiple R-squared:  0.8554, Adjusted R-squared:  0.7448
## F-statistic: 7.734 on 13 and 17 DF,  p-value: 8.654e-05
```

Nous remarquons que **X** devient plus significative avec une $p-value$ faible 0.00934 et **U2** a encore significativit? int?ressante.

N.B. : l'int?r?t de la variable **Intercept** se r?sume ? de l'overfiting des donn?es, ce qui n'est pas exploitable en terme de pr?diction.

Nous utilisons notre mod?le sur les donn?es de test :

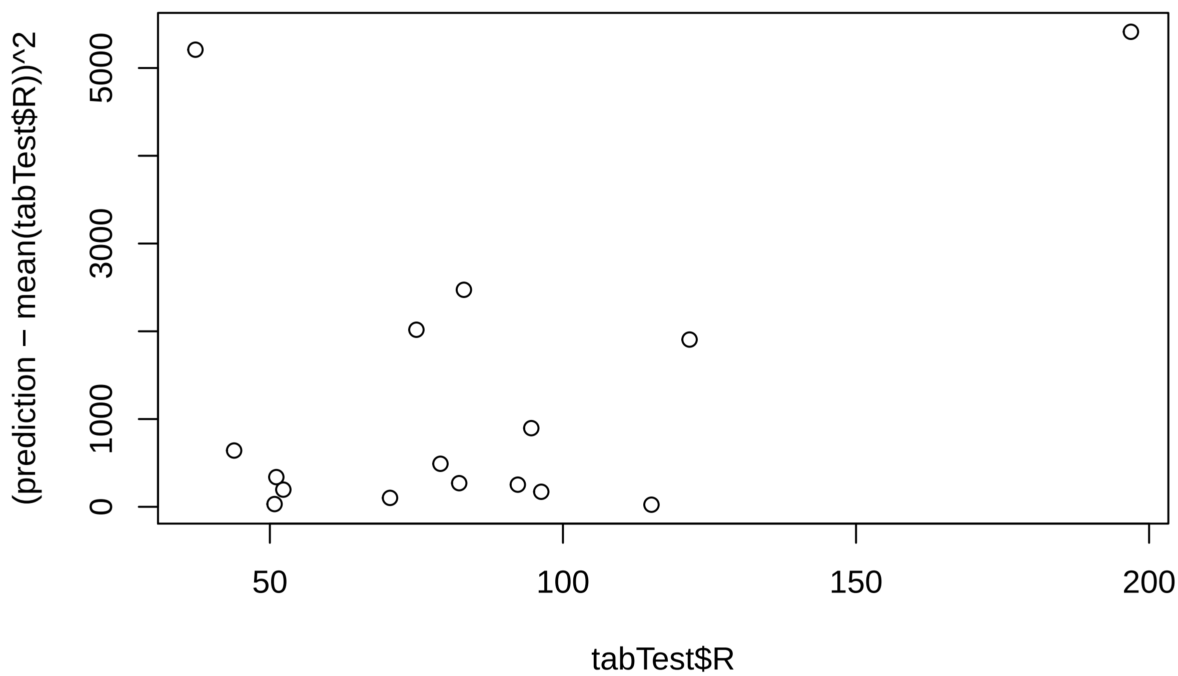
```
prediction = predict(res,tabTest)

mean((prediction-mean(tabTest$R))^2)
```

```
## [1] 1276.863
```

```
sqrt(var((prediction-mean(tabTest$R))^2))
```

```
## [1] 1750.151
```

```
plot(tabTest$R,(prediction-mean(tabTest$R))^2)
```

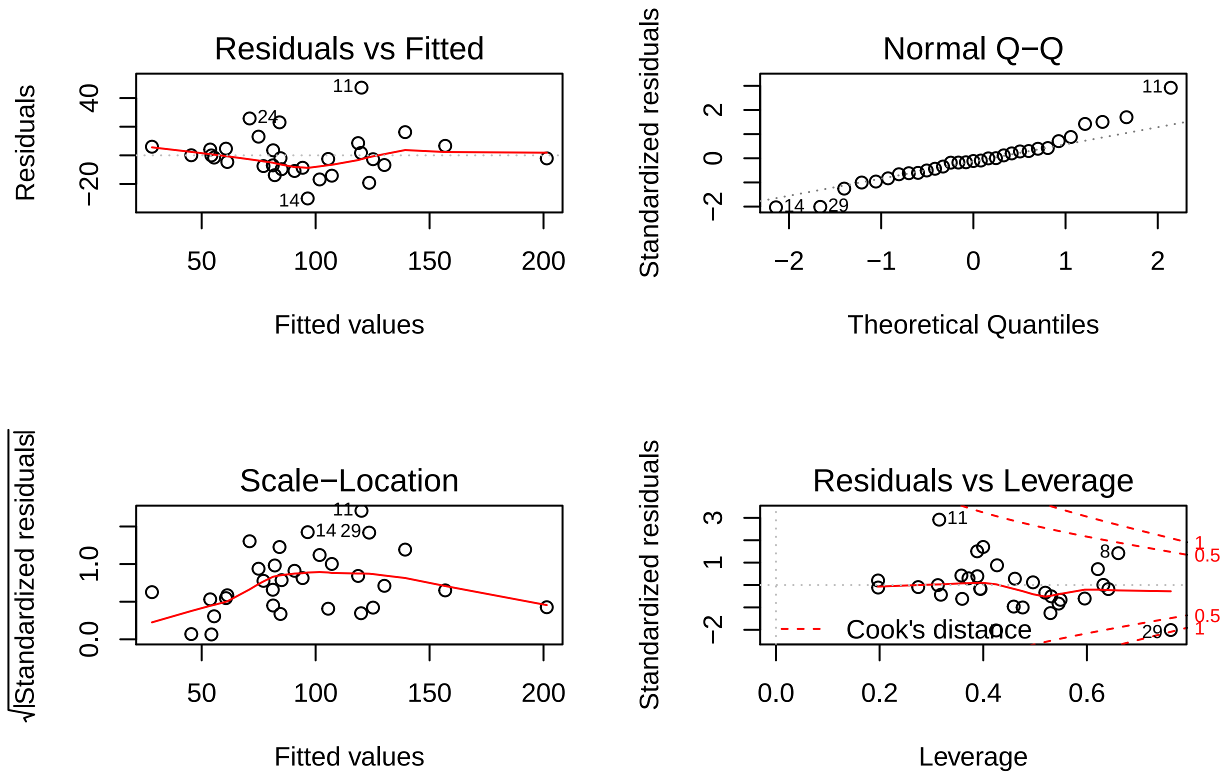


La moyenne des erreurs est ? 1276.863 et l'?cart-type et ? 1750.151. En conclusion, les moyennes des erreurs sont ?lev?es pour ce mod?le. Il n'est pas suffisament pertinant.

7. Analyse graphique

Par analyse graphique :

```
x11()
par(mfrow=c(2,2))
plot(res)
```



On observe que l'allure des r?sidus suit tr?s relativement une loi normale. Notamment nous constatons dans *Residuals vs Fitted* et *Scale-Location* que la loi n'est pas exactement centr?e en z?ro. Ensuite dans *Residuals vs Leverage* nous constatons que les valeurs des r?sidus ne sont pas trop grandes. Enfin dans *Normal Q-Q*, nous constatons que la loi n'est pas normale aux extr?mit?s.

Application II : m?lange de classifieurs, algorithme EM

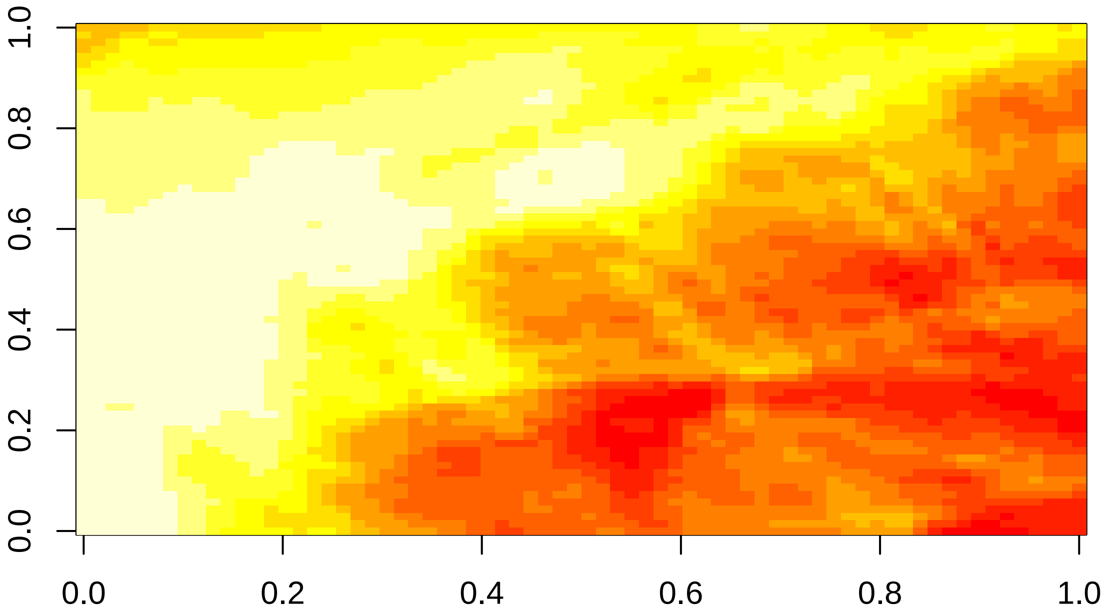
M?lange de gaussiennes

Chargement du fichier *irm_thorax.txt* :

```
irm=as.matrix(read.table("irm_thorax.txt",header=F,sep=';'))
```

G?n?ration de l'image :

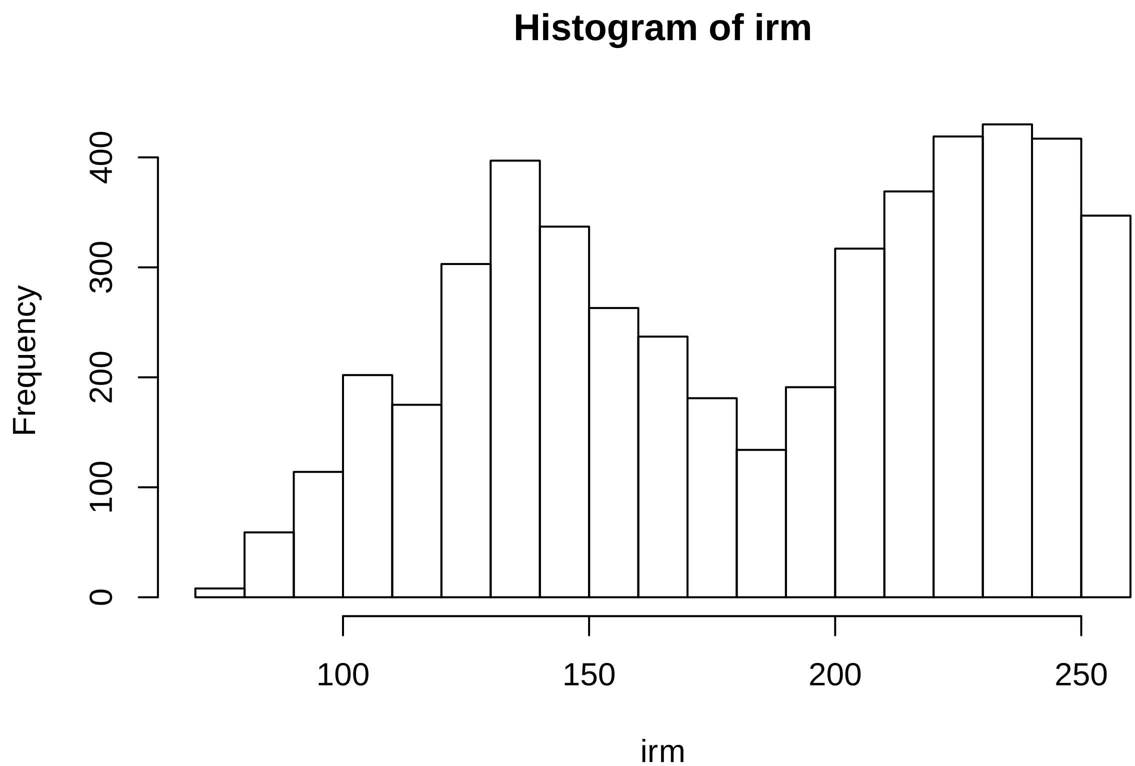
```
image(irm)
```



Nous observons une fronti?re relativement oblique entre deux zones color?es: rouge en bas-droite et jaune en haut-gauche. Nous devinons la colonne vert?brale d'un homo-sapiens.

Affichage de l'histogramme :

```
hist(irm)
```



Nous observons que les donn?es sont distribu?es selon 2 lois gaussiennes de moyennes qui semblent ?tre 135 et 240.

3 & 4. Impl?mentation de l'algorithme Expectation-Maximization

Nous avons impl?ment? un algorhithme EM qui prend en param?tre les donn?es et le nombre de m?lange de K loi gaussienne.

1) Initialisation des vecteurs et matrices

2) It?ration jusqu'? convergence

3) E-Step

Etape **E**: calcul de $t_i k$ par la r?gle d'inversion de Bayes:

$$t_{ik} = \frac{\pi_k^{(c)} f(x_i, \theta_k^{(c)})}{\sum_{\ell=1}^{g} \pi_\ell^{(c)} f(x_i, \theta_\ell^{(c)})}$$

Dans le cas du m?lange gaussien avec K=2 et d=1 nous calculons les responsabilit?s tel que:

$$\widehat{\eta}^{(t+1)} = \frac{p_1^{(t)} f_{\mu_1^{(t)}, \Sigma_1^{(t)}}(x_i)}{f_{X|p^{(t)}, \theta^{(t)}(x_i)}}$$

avec,

$$p_1^{(t+1)} = \frac{1}{n} \Sigma_{i=1}^{n} \widehat{\eta}^{(t+1)}$$

4) M-Step

?tape **M**: d?termination de Φ maximisant

$$Q\left(\Phi, \Phi^{(c)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{g} t_{ik} \log\left(\pi_k f(x_i, \theta_k)\right)$$

Les proportions optimales sont donn?es par:

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

Dans le cas du m?lange gaussien avec K=2 et d=1 nous calculons les moyennes/variances:

$$\mu_1^{(t+1)} = \Sigma_{i=1}^{n} \frac{\widehat{\eta}^{(t+1)}}{\Sigma_{i=1}^{n} \widehat{\eta}^{(t+1)}} x_i$$

et

$$(\sigma_1^{(t+1)})^2 = \Sigma_{i=1}^{n} \frac{\widehat{\eta}^{(t+1)}}{\Sigma_{i=1}^{n} \widehat{\eta}^{(t+1)}} (x_i - \mu_1^{(t+1)})^2$$

Notre impl?mentation de l'algorithme EM :

```
algoEMgaussien = function(data, K) {
  dataSize = length(data)
  # On d?finit les probabilit?s p(i), moyenne mu et les variances des gaussiennes.
  p = c()
  mu = c()
  sigma = c()
  eta = matrix(nrow=dataSize, ncol=K) #responsabilit?s de chaque point

  # Initialisation des donn?es.
  for (i in 1:K)
  {
    p[i]=1/K
    mu[i]=data[sample(2:dataSize, 1)] # au hasard dans data
    sigma[i]=sqrt(var(data))
  }

  #Initialisation des clusters
  clusters = rep(1:dataSize)
  #Variables de condition d'arret
  newSum=0
  oldSum=0
  iterationNumber = 0
  while(TRUE){
    #E
    for (i in 1:K)
    {
      eta[,i]=p[i]*dnorm(data, mean=mu[i], sd=sigma[i])
    }

     for (l in 1:dataSize)
    {
      clusters[l] = which.max(eta[l,]) #le max du vecteur eta
```

```r
      eta[l,] = eta[l,]/sum(eta[l,])
     }

    # P(Y=k)
    for (i in 1:K)
    {
      cluster = which(clusters == i)
      p[i]=length(cluster)/dataSize
    }

    #M step
    for (i in 1:K)
    {
      cluster = which(clusters == i)
      #actualisation des moyennes et des deviations
      mu[i]=sum(data[cluster])/length(cluster)
      sigma[i]=sqrt(var(data[cluster]))
    }

    oldSUm = newSum
    newSum = 0
    for (i in 1:K)
    {
      cluster = which(clusters == i)
      newSum = newSum + sum(log(p[i]*dnorm(data[cluster],mu[i],sigma[i])))
    }
    iterationNumber=iterationNumber+1

    #message("Iteration ", iterationNumber)
    #Si un des sigma est proche de z?ro alors nous risquons d'avoir une donn?e par cluster.
   if(min(sigma)<0.1){
      message("Overfitting des donn?es ! Variance->0 ", min(sigma))
      break
   }
   if(iterationNumber>100 || abs((newSum-oldSUm)/newSum) <0.000000001 )
    {
      break
    }
  }
  #On retourne un affichage en liste des probabilit?s, esp?rances, ?carts-type de chacune des gaussienn
  return (list("Probabilit?s des K gaussiennes: "=p, "Esp?rances des K gaussiennes: "=mu, "Sigma des K g
}

result=algoEMgaussien(as.vector(irm), 2)
print(result)
```

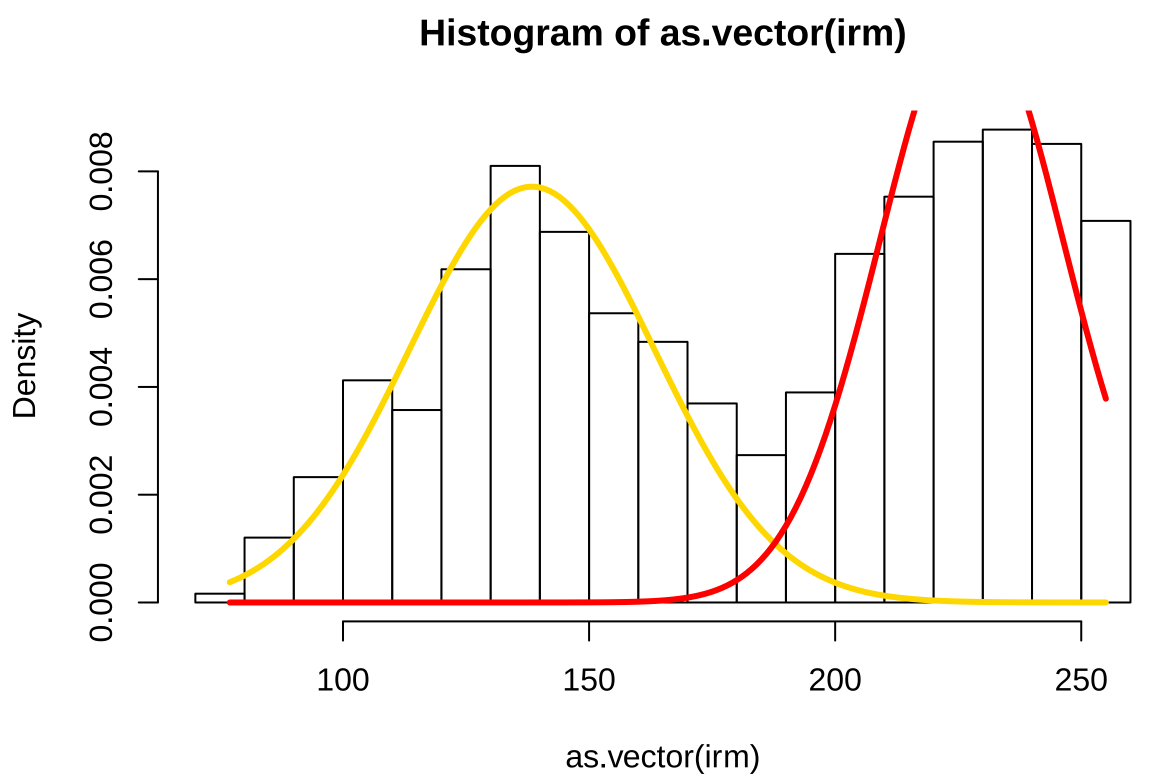
```
## $`Probabilit?s des K gaussiennes: `
## [1] 0.4830612 0.5169388
##
## $`Esp?rances des K gaussiennes: `
## [1] 138.3899 227.7154
##
## $`Sigma des K gaussiennes`
```

```
## [1] 24.96724 18.61188
##
## $`Nombre d'iterations`
## [1] 9
```

Les probabilit?s obtenues pour K=2 sont proches de 50%. On a : $p_1 = 0.483$ et $p_2 = 0.517$. Les moyennes sont proches des observations $mu_1 = 138.39$ et $mu_2 = 227.71$ et les ?carts-type sont $sigma_1 = 24.97$ et $sigma_2 = 18.61$. Ces r?sultats confirment la conjecture de nos observations pr?c?dentes.

Histogramme avec nos K=2 gaussiennes :

```
hist(as.vector(irm), prob=TRUE)
x = seq(min(as.vector(irm)), max(as.vector(irm)))

k1 = 0.483*dnorm(x, mean=138.39, sd=24.97)
lines(x, k1, col="gold", lwd = 3);

k2 = 0.517*dnorm(x, mean=227.71, sd=18.61)
lines(x, k2, col="red", lwd = 3);
```



Pour des K plus grand on observe un ph?nom?ne d'overfit des donn?es.

```
print(algoEMgaussien(as.vector(irm), 3))
```

```
## $`Probabilit?s des K gaussiennes: `
## [1] 0.39795918 0.52959184 0.07244898
```

```
##
## $`Esp?rances des K gaussiennes: `
## [1] 130.6938 226.6929 172.5380
##
## $`Sigma des K gaussiennes`
## [1] 20.283120 19.516557  5.110153
##
## $`Nombre d'iterations`
## [1] 11
```

```r
print(algoEMgaussien(as.vector(irm), 5))
```

```
## $`Probabilit?s des K gaussiennes: `
## [1] 0.13959184 0.04897959 0.18163265 0.52428571 0.10551020
##
## $`Esp?rances des K gaussiennes: `
## [1] 166.5716 123.6083 139.3315 227.1296 103.0368
##
## $`Sigma des K gaussiennes`
## [1]  8.938171  2.652214  6.920342 19.123388 10.046240
##
## $`Nombre d'iterations`
## [1] 21
```

En conslusion l'utilisation de K=2 gaussienne est pertinante pour le cas ?tudi?.

Mixture de r?gressions par l'algorithme EM

5. Chargement de la biblioth?que

Installation et chargement de mixtools :

```r
library(mixtools)
```

```
## mixtools package, version 1.0.4, Released 2016-01-11
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
```

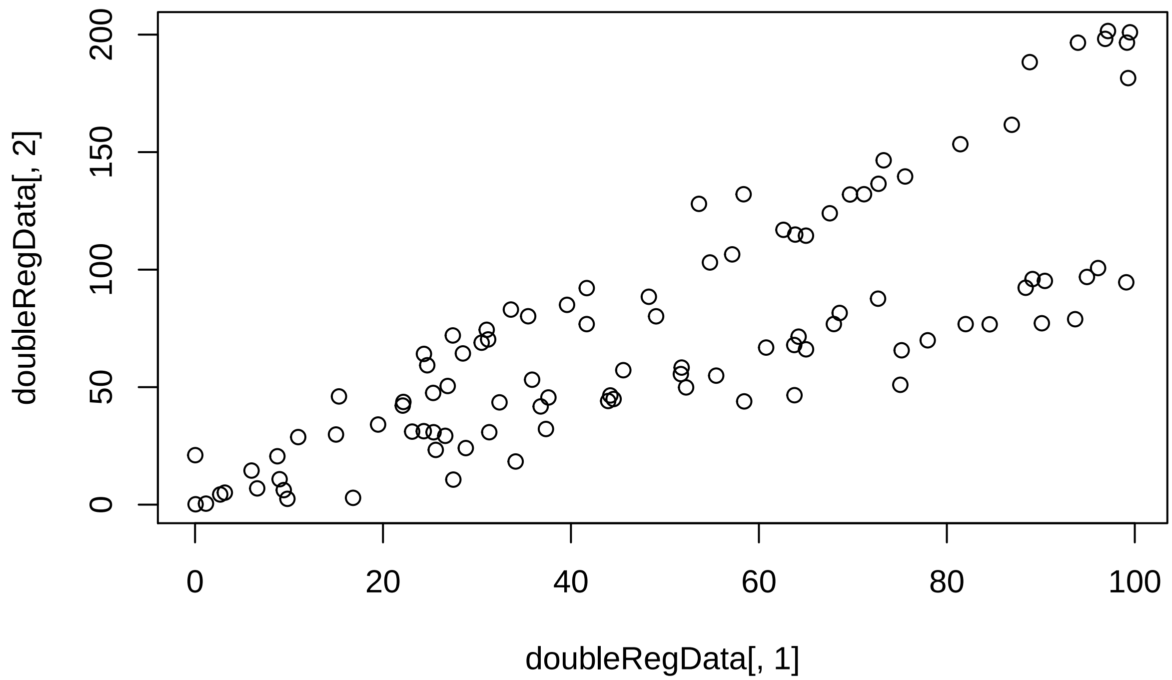
6. Importation des donn?es

Import des donn?es et affichage :

```r
tab2 = read.table("regression_double.txt", sep=";")
summary(tab2)
```

```
##        V1                 V2
##  Min.   : 0.02142   Min.   :  0.1681
##  1st Qu.:26.37116   1st Qu.: 39.8648
##  Median :46.92929   Median : 66.4646
##  Mean   :49.44388   Mean   : 73.2069
##  3rd Qu.:72.69059   3rd Qu.: 95.4044
##  Max.   :99.48837   Max.   :201.5280
```

```r
doubleRegData = as.matrix(tab2)
plot(doubleRegData[,1],doubleRegData[,2])
```



On observe que les donn?es sont r?parties selon deux droites. On en d?duit qu'une r?gression lin?aire simple ne conviendra pas. On peut en conclure qu'une mixture de K=2 gaussiennes serait un choix judicieux.

7. Mix EM

R?alisation d'un Mix EM:

```r
mixmodel = regmixEM(doubleRegData[,1], doubleRegData[,2], k=2)
```

```
## number of iterations= 34
```

8. Affichage du r?sultat et calcul des r?sidus

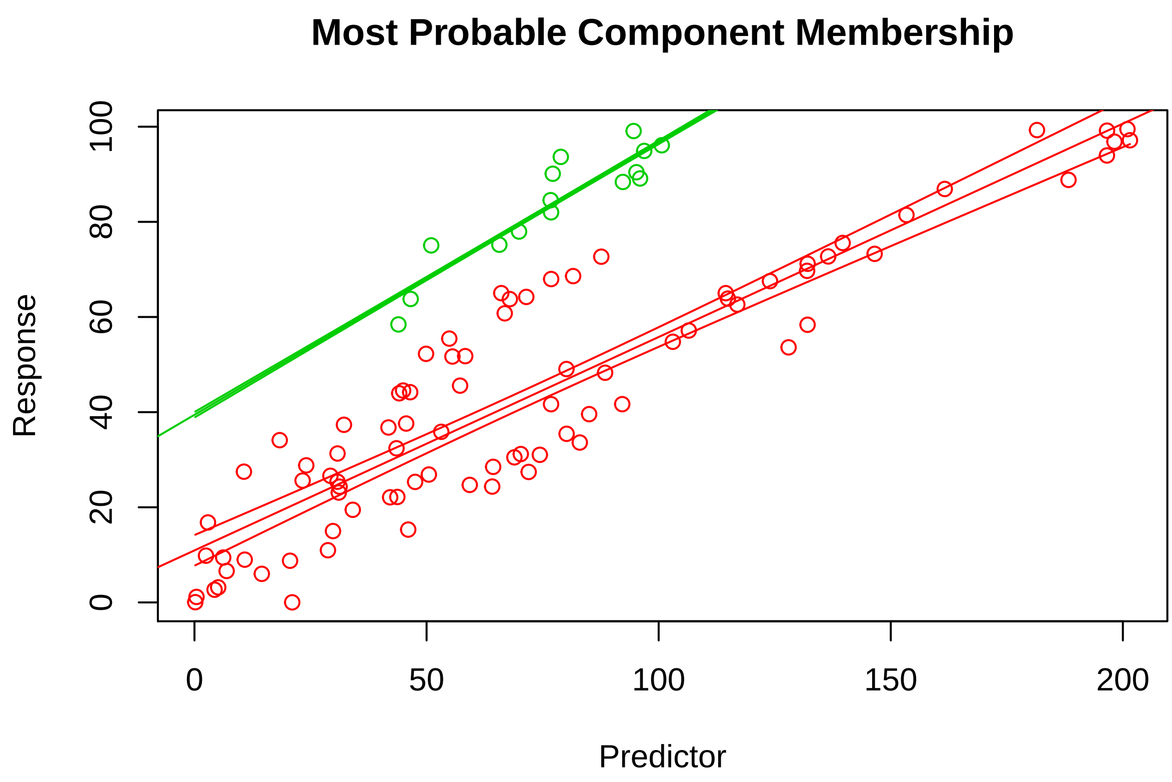
R?sultat du Mix EM avec K=2:

```r
summary(mixmodel)
```

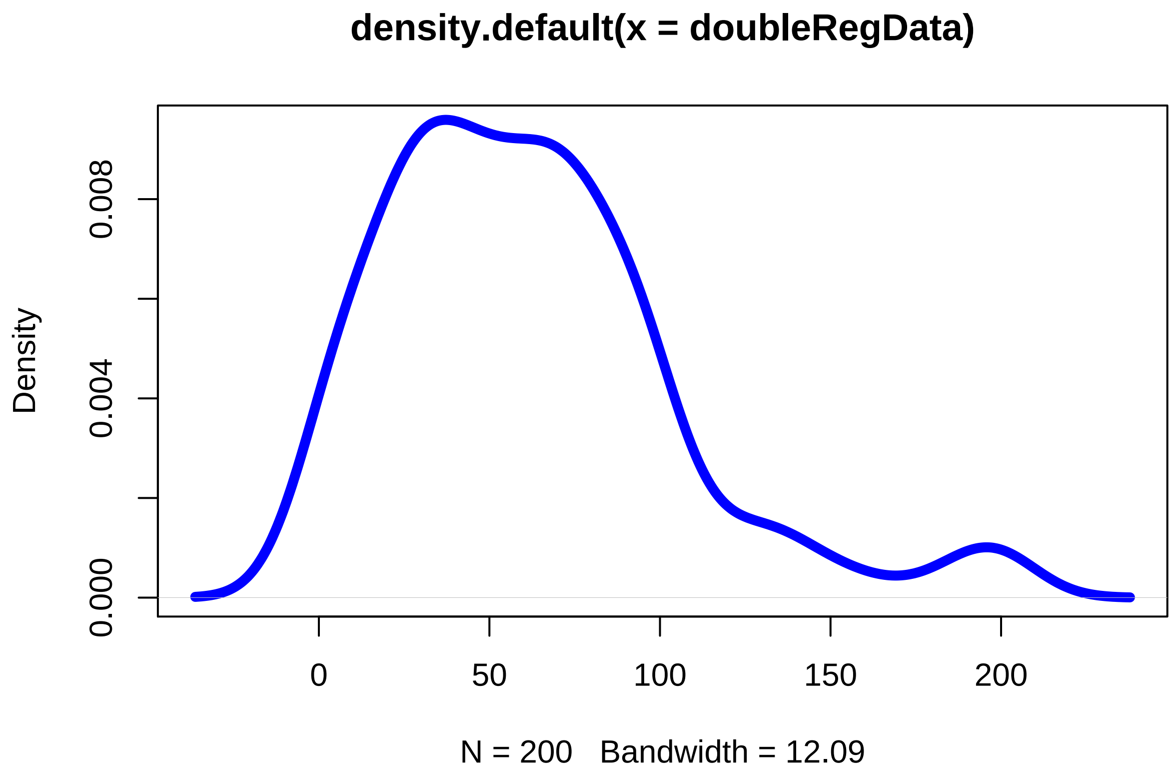
```
## summary of regmixEM object:
##             comp 1    comp 2
## lambda   0.851548  0.148452
```

```
## sigma  11.130945  4.747917
## beta1  10.940031 39.442482
## beta2   0.448559  0.572777
## loglik at estimate:  -409.9987
```

```r
plot(mixmodel, which=2)
```




```r
plot(density(doubleRegData), col='blue', ylim=c(0,0.0095), lwd=5)
```



Au bout d'une quizaine d'it?rations nous obtenons une mixture de deux gaussiennes. Ce qui confirme nos attentes.

Calcul des r?sidus:

```
print(mixmodel["sigma"][1])
```

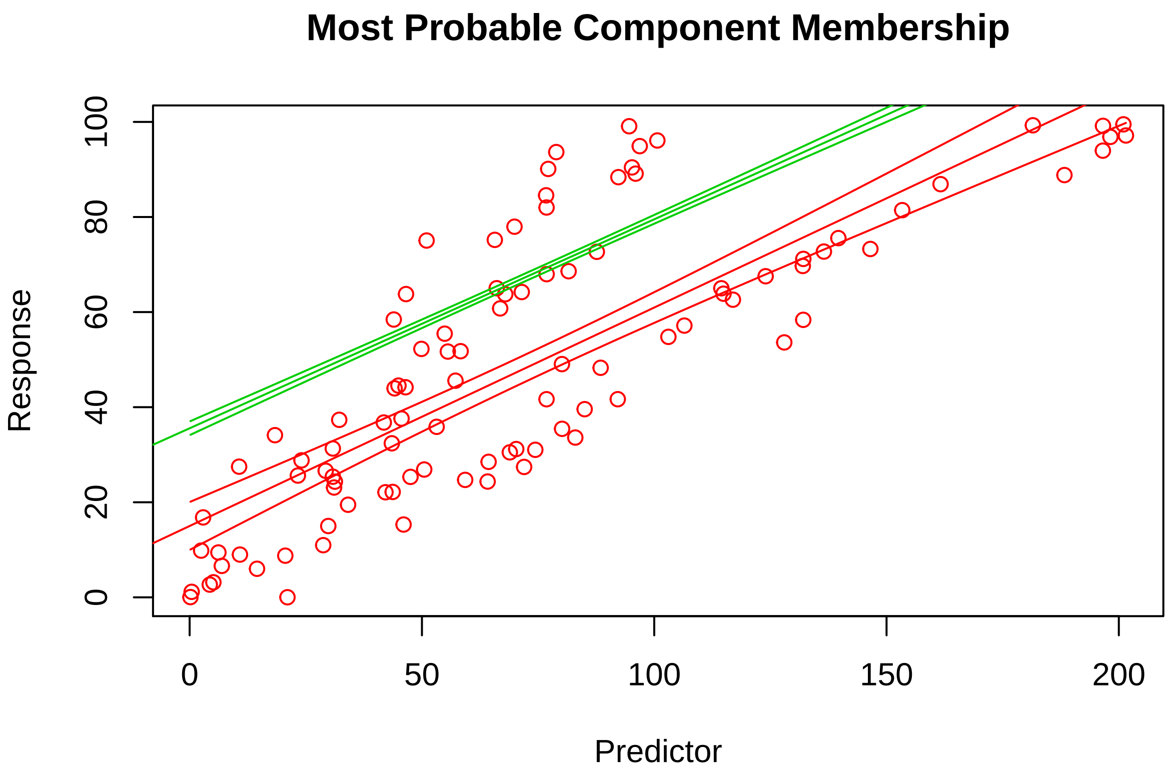
```
## $sigma
## [1] 11.130945  4.747917
```

9. Mix EM sur 1,3 et 5 it?rations

```
regMix1 = regmixEM(doubleRegData[,1], doubleRegData[,2], k= 2, maxit=1)
```

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1
```

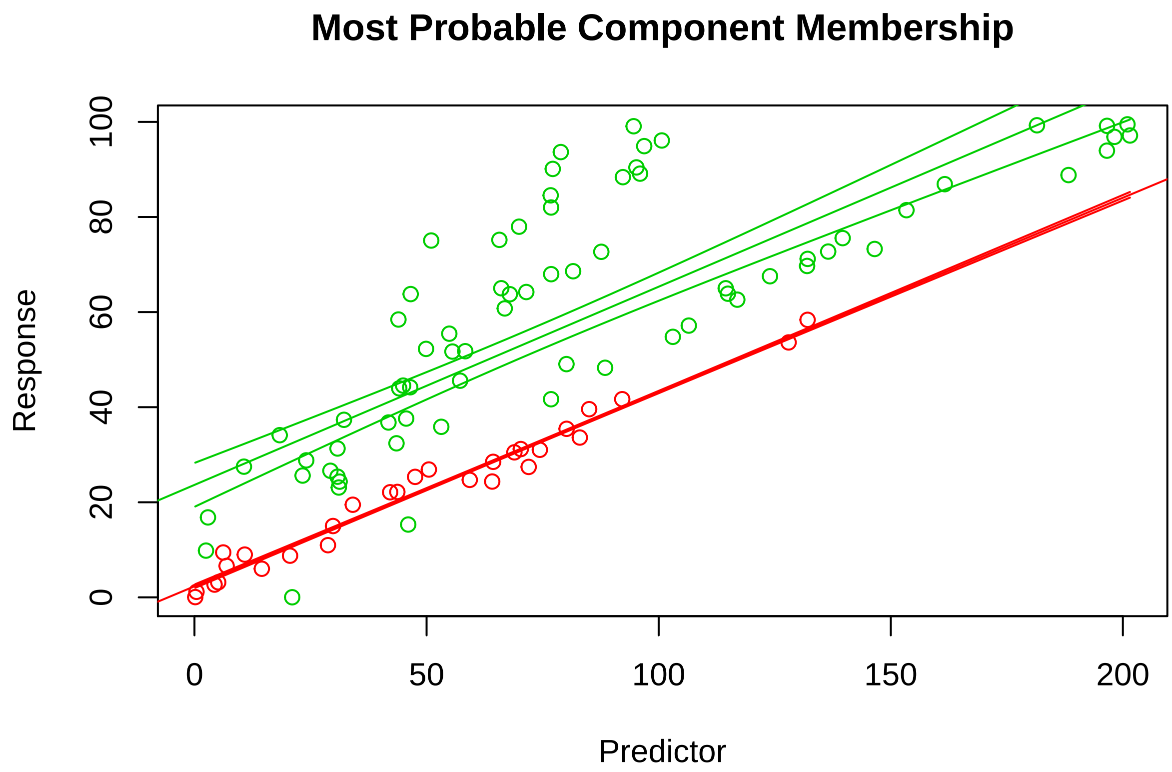
```
plot(regMix1, which=2)
```




```
regMix2 = regmixEM(doubleRegData[,1], doubleRegData[,2], k= 2, maxit=3)
```

```
## WARNING! NOT CONVERGENT!
## number of iterations= 3
```

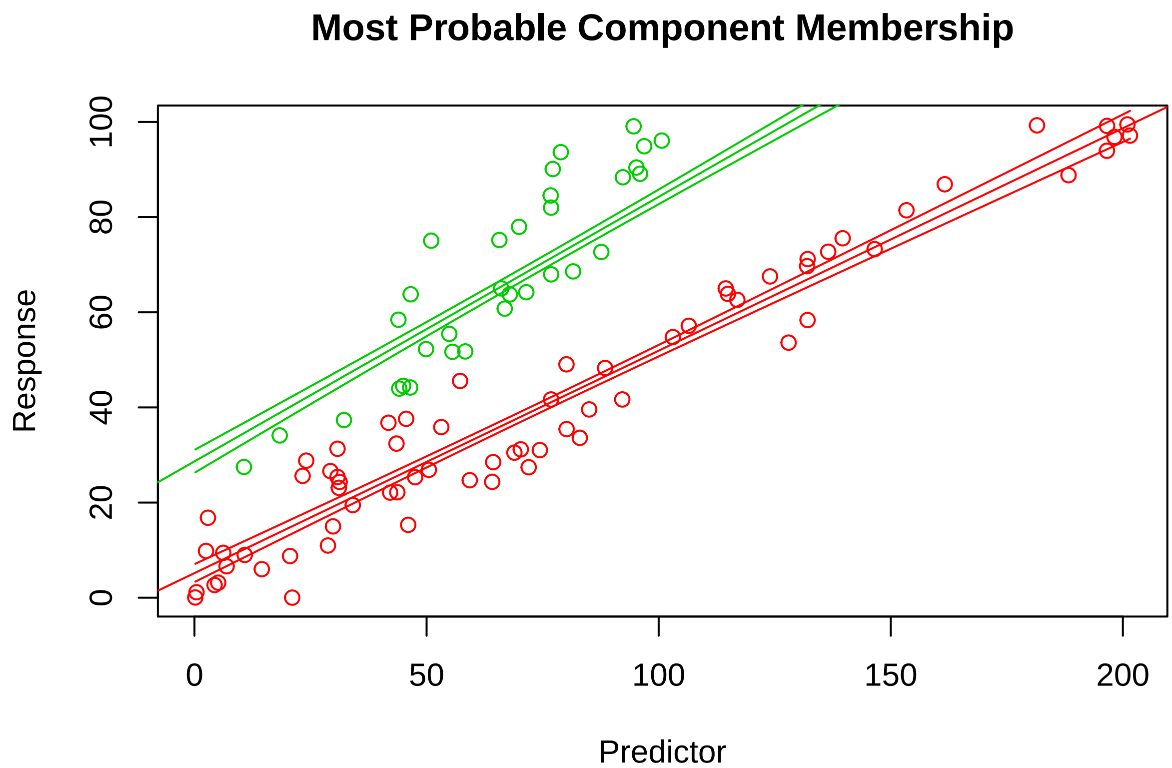
```
plot(regMix2, which=2)
```




```r
regMix3 = regmixEM(doubleRegData[,1], doubleRegData[,2], k= 2, maxit=5)
```

```
## WARNING! NOT CONVERGENT!
## number of iterations= 5
```

```r
plot(regMix3, which=2)
```



Apr?s 1, 3 et 5 it?rations nous remarquons que l'on tend vers le mod?le, cependant le r?sultat pour un nombre faible d'it?rations est influenc? par la fa?on dont sont initialis?es les moyennes et les ?carts-type dans l'algorithme regmixEM de la biblioth?que mixtools.