Clustering using the kmeans algorithm

Bealy MECH

10/28/2021

Exercice 1 : Partition and Matrix

Write the R code which produces the partition matrix

1st solution

```
data(iris)
dim(iris)
```

```
## [1] 150   5
```

```
X <- iris[, 1:4] # we don't consider the last column which is "Species"
library(nnet)
# la matrice de la partition
C <- class.ind(iris$Species)
```

La matrice C renvoit 1 si la ligne est dans l'espace (same species), O sinon.

```
summary(iris$Species)  # There are 3 species, normally
```

```
##     setosa versicolor  virginica
##         50         50         50
```

```
t(X) %*% C
```

```
##              setosa versicolor virginica
## Sepal.Length  250.3      296.8     329.4
## Sepal.Width   171.4      138.5     148.7
## Petal.Length   73.1      213.0     277.6
## Petal.Width    12.3       66.3     101.3
```

```
diag(t(C) %*% C)   # return the number of each species
```

```
##     setosa versicolor  virginica
##         50         50         50
```

```r
# the matrix of gravity centers of the quantitative variables
t((t(X) %*% C)/diag(t(C) %*% C))
```

```
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```

Remarque

- $t(C)*C$ renvoie une matrice diagonale donnant le nombre dans chaque espece, et donc $diagonal(t(C)*C)$ renvoie un vecteur composé de la diagonale.
- La dernière matrice donne la moyenne par espece et par variable.

2nd solution

```r
X <- iris[, 1:4]
dim(X)
```

```
## [1] 150   4
```

```r
names(X)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
```

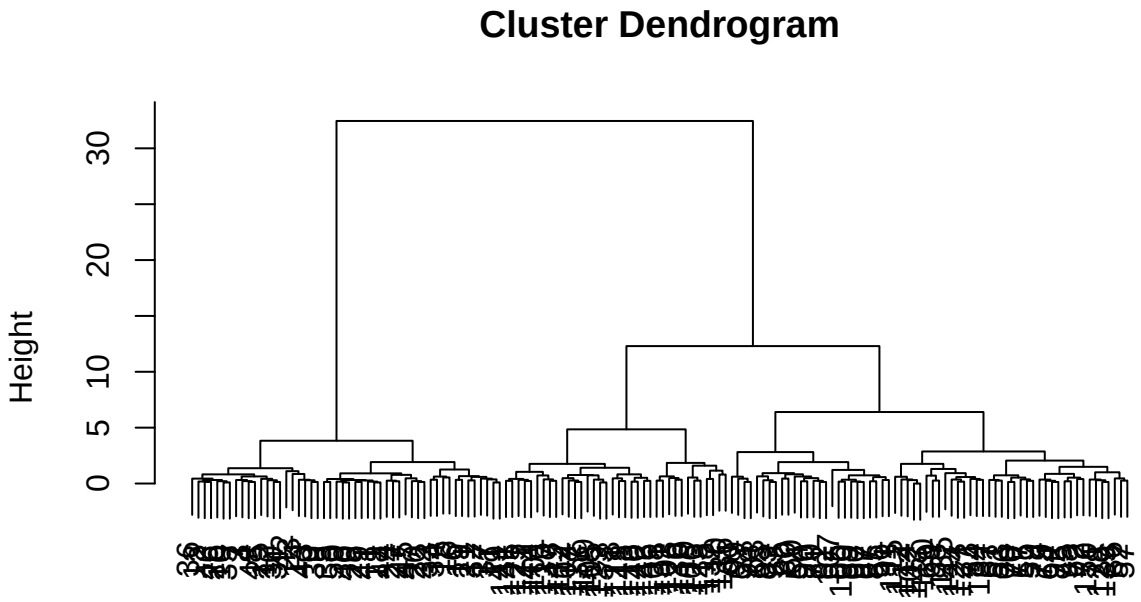
```r
cluster <- iris$Species
# Compute the matrix C = (Cik) where Cik=1 if i belongs to Ck, 0 otherwise
C <- matrix(0, 150, 3)  # matrix 0 composes of 150 rows and 3 columns
for (i in seq(1, 150)) { # from 1 to 150 increase by 1
  if(cluster[i] == "setosa") {
    C[1, 1] = 1  # first column is the data of species "setosa"
  }
  if(cluster[i] == "versicolor") {
    C[i, 2] = 1  # second column is the data of species "versicolor"
  }
  if(cluster[i] == "virginica") {
    C[i, 3] = 1  # third column is the data of species "virginica"
  }
}
# verify the matrix C
```

Compute M which represents the matrix of gravity center.

```r
X <- as.matrix(X)
M <- solve(t(C)%*%C)%*%t(C)%*%X
M
```

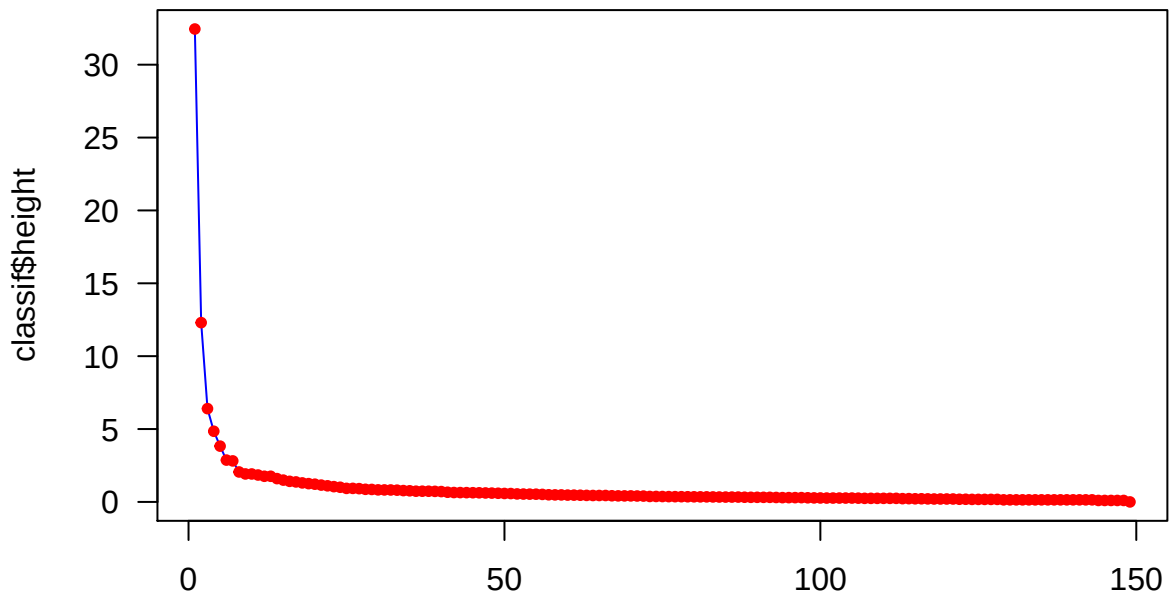
```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]        5.100       3.500        1.400       0.200
## [2,]        5.936       2.770        4.260       1.326
## [3,]        6.588       2.974        5.552       2.026
```

```
dX <- dist(X)
classif <- hclust(d = dX, method = "ward.D2")
plot(classif)
```



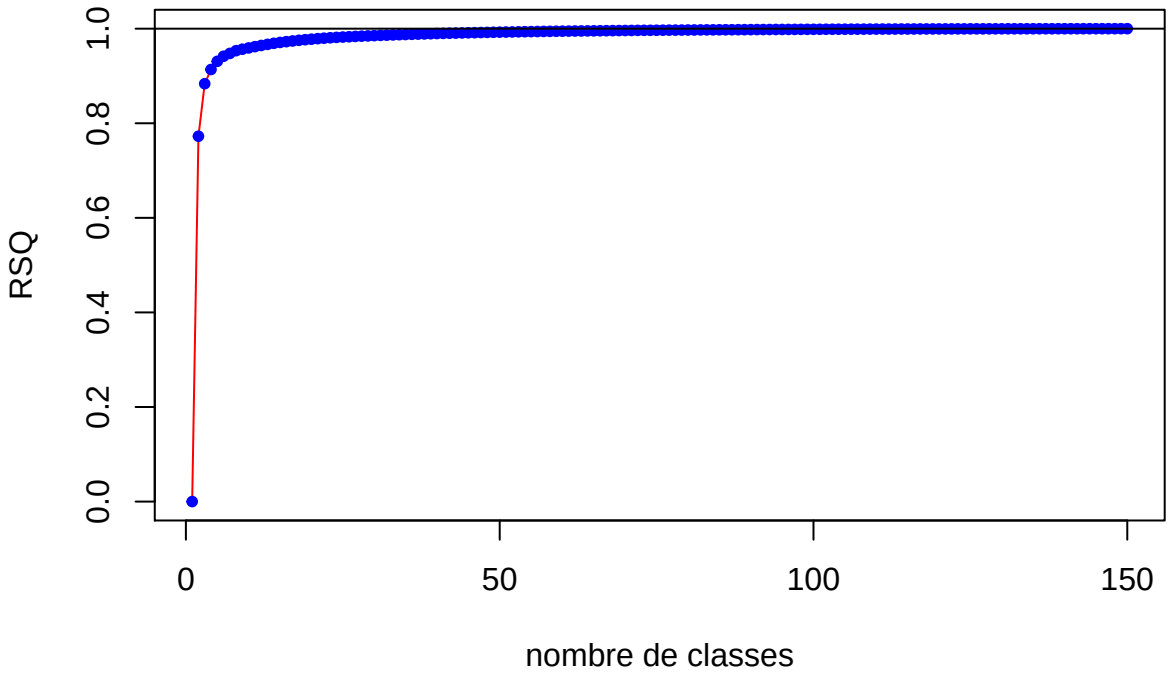

```
plot(rev(classif$height), type='l', main="hauteurs du dendrogramme décroissantes", ylab="classif$height

points(1:length(classif$height), rev(classif$height), pch=20, col="red")
```




```
c <- (cbind(1:length(classif$height), rev(classif$height)))
```

```
c2 <- diff(c[,2])
res <- diff(c2)
P <- X
RSQ <- rep(0, nrow(P))
SQTot <- sum(scale(P,scale=FALSE)^2)
for (i in 1:nrow(P)){
  Cla <- as.factor(cutree(hclust(dX,"ward.D2"),i))
  RSQ[i] <- sum(t((t(sapply(1:ncol(P),function(i) tapply(P[,i],Cla,mean)))-apply(P,2,mean))^2)*as.vecto
}
plot(RSQ, type='l', col="red",xlab="nombre de classes")
points(1:length(RSQ),RSQ,pch=20, col="blue")
abline(h=1)
```



Exercise 2 : The Bell Number

1. Show that the number of partition of n objects verifies:

$$B_{n+1} = \sum_{k=0}^{n} B_k$$

Cette formule s'appelle également **la relation d'Aitken**.

Exercise 4 : Clustering of the crabs (library MASS)

```
library(MASS)
data(crabs)
dim(crabs)
```

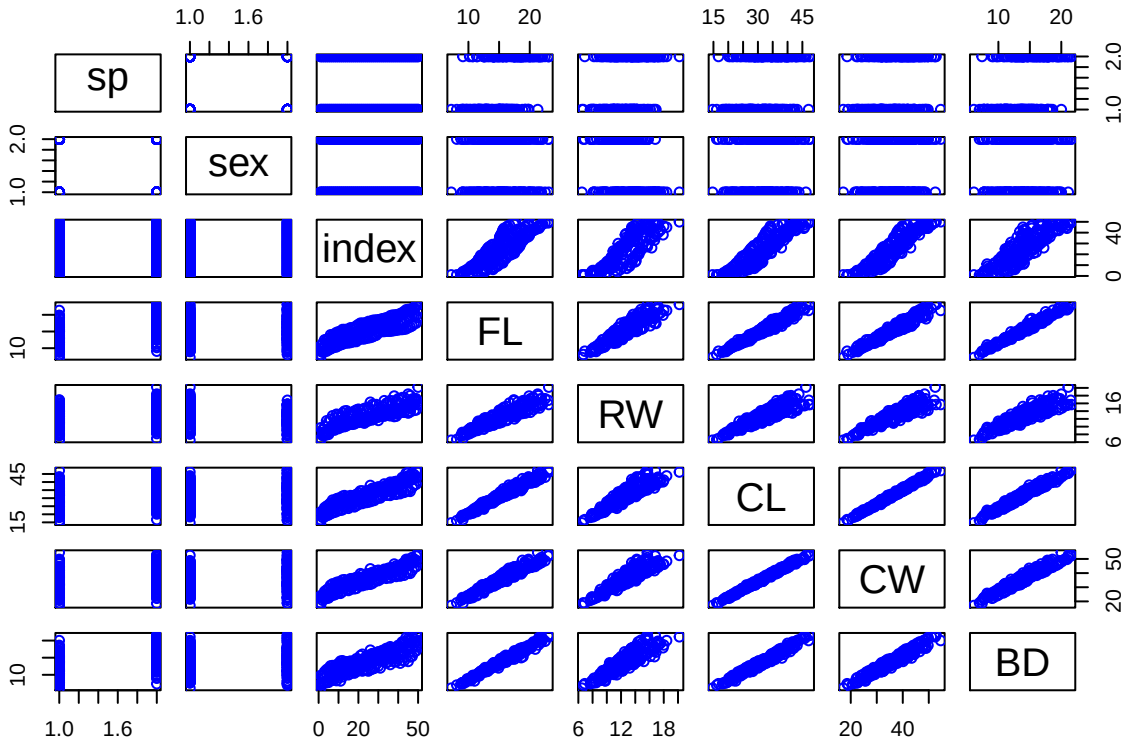
```
## [1] 200   8
```

```
names(crabs)
```

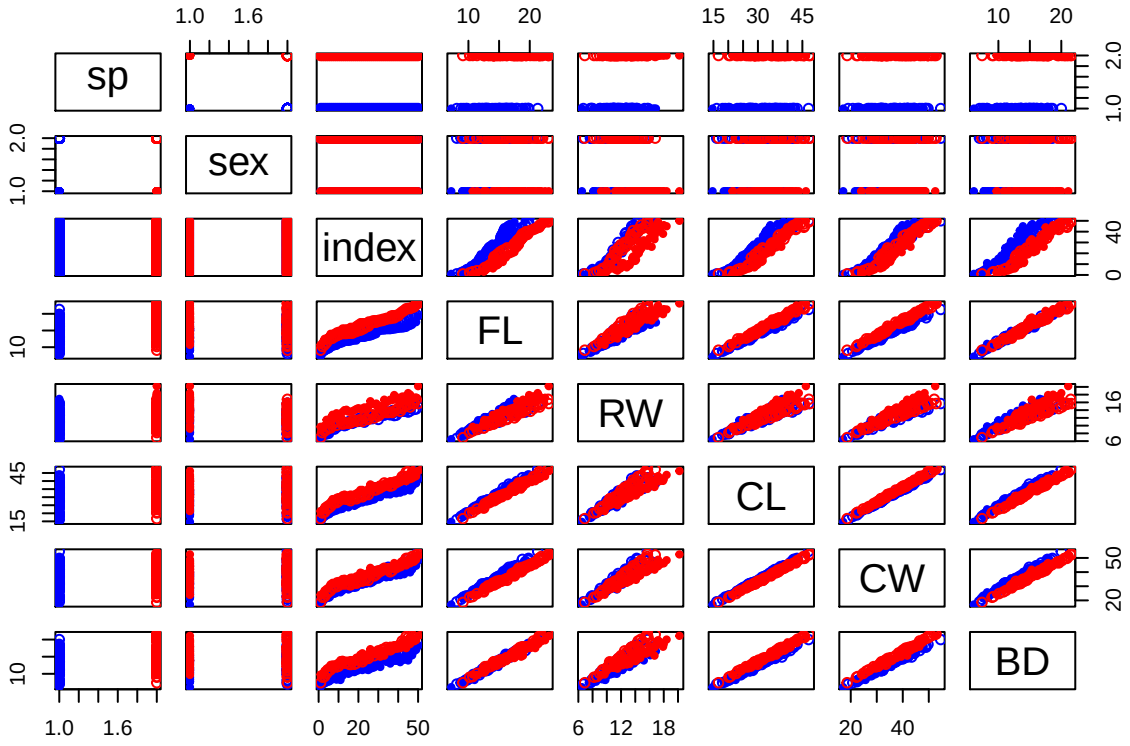
```
## [1] "sp"    "sex"   "index" "FL"    "RW"    "CL"    "CW"    "BD"
```

Represent all crabs with a color corresponding to the specy and symbol to the sex

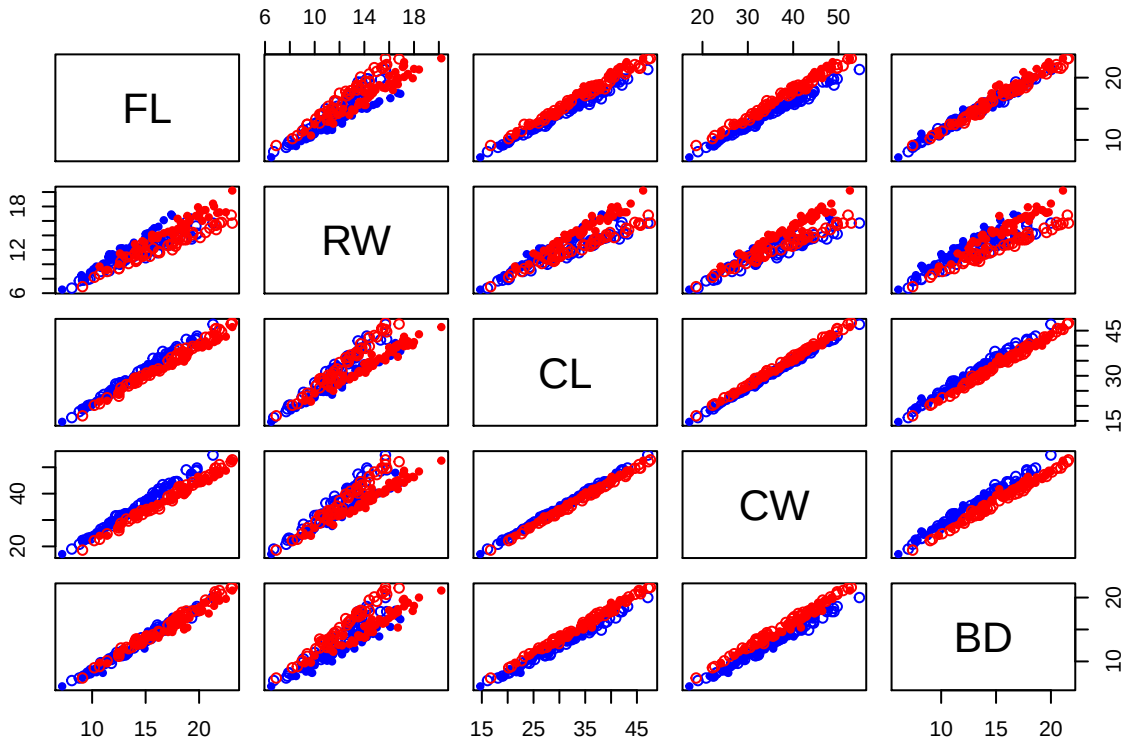
```
plot(crabs, col = "blue")
```




```
pairs(crabs, col = c("blue", "red")[crabs$sp], pch = c(20, 21)[crabs$sex])
```




```
crabsquant <- crabs[, 4:8]
pairs(crabsquant, col = c("blue", "red")[crabs$sp], pch = c(20, 21)[crabs$sex])
```

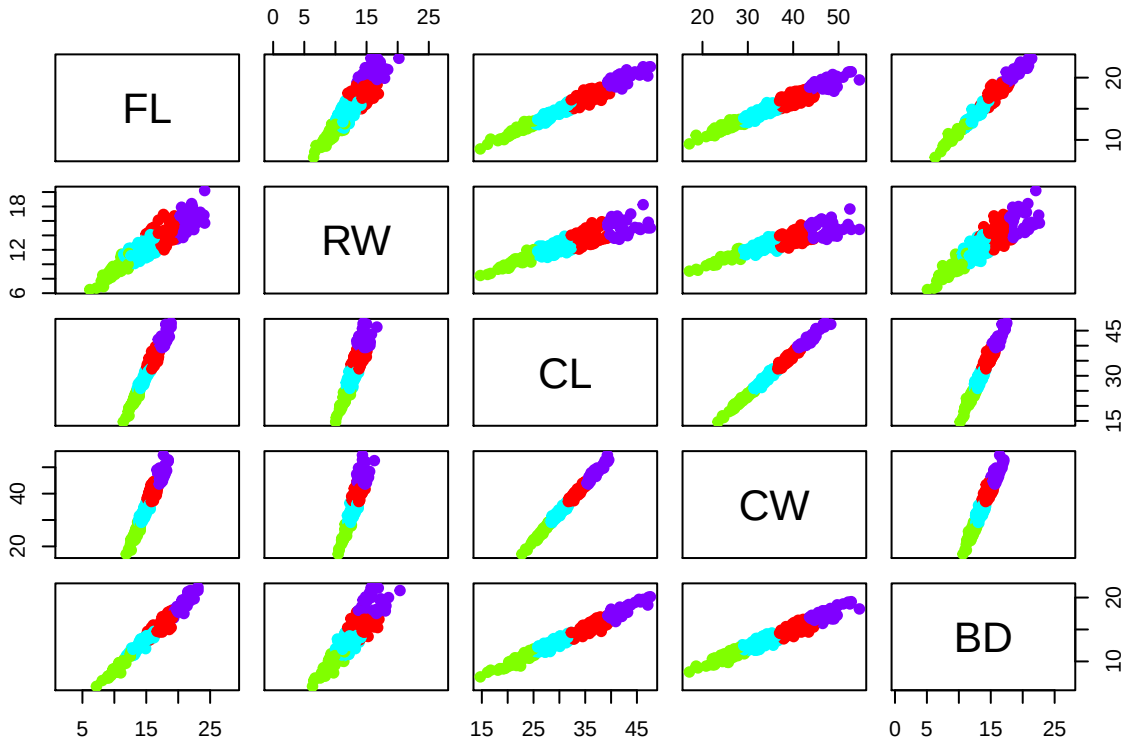



```
# set a color for each specy and a different symbol per sex
summary(crabs$sex)  # Male and Female
```

```
##   F   M
## 100 100
```

Use kmeans to find 4 classes in crabsquant

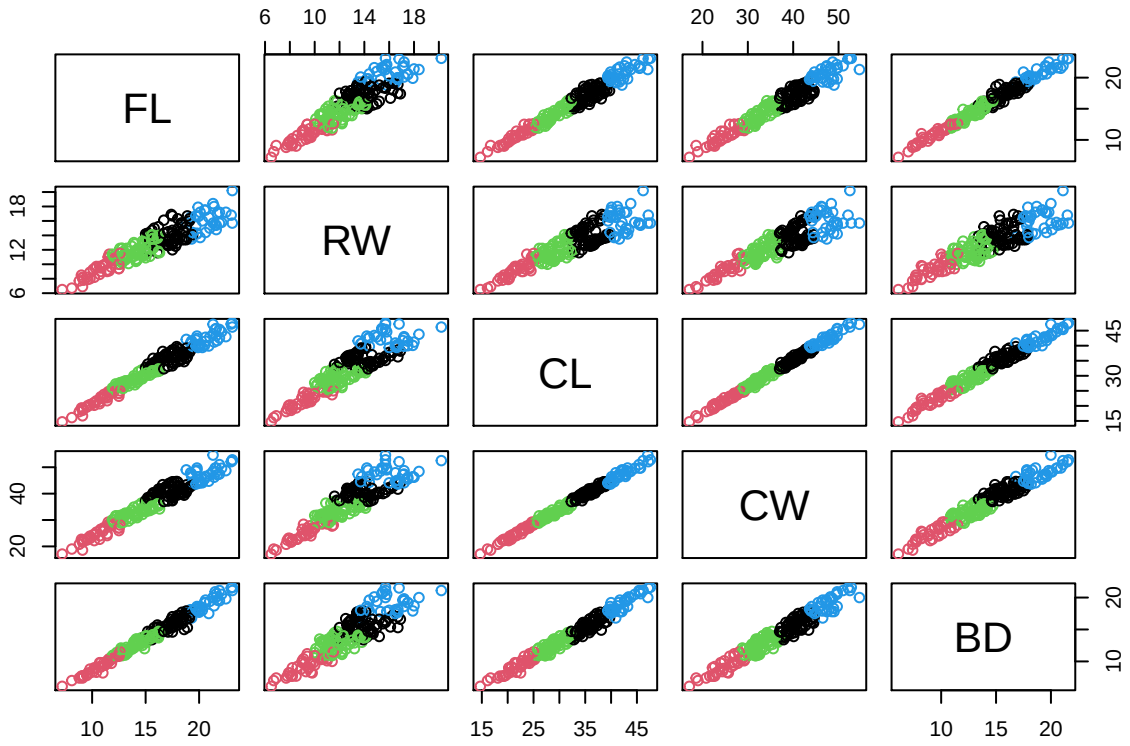
```
nbclasse <- 4
KM <- kmeans(crabsquant, nbclasse)
plot(crabsquant, asp = 1, pch = 19, col = rainbow(nbclasse)[KM$cluster])
points(KM$centers, pch = 8, col = rainbow(nbclasse)[KM$cluster], cex = 2)
```



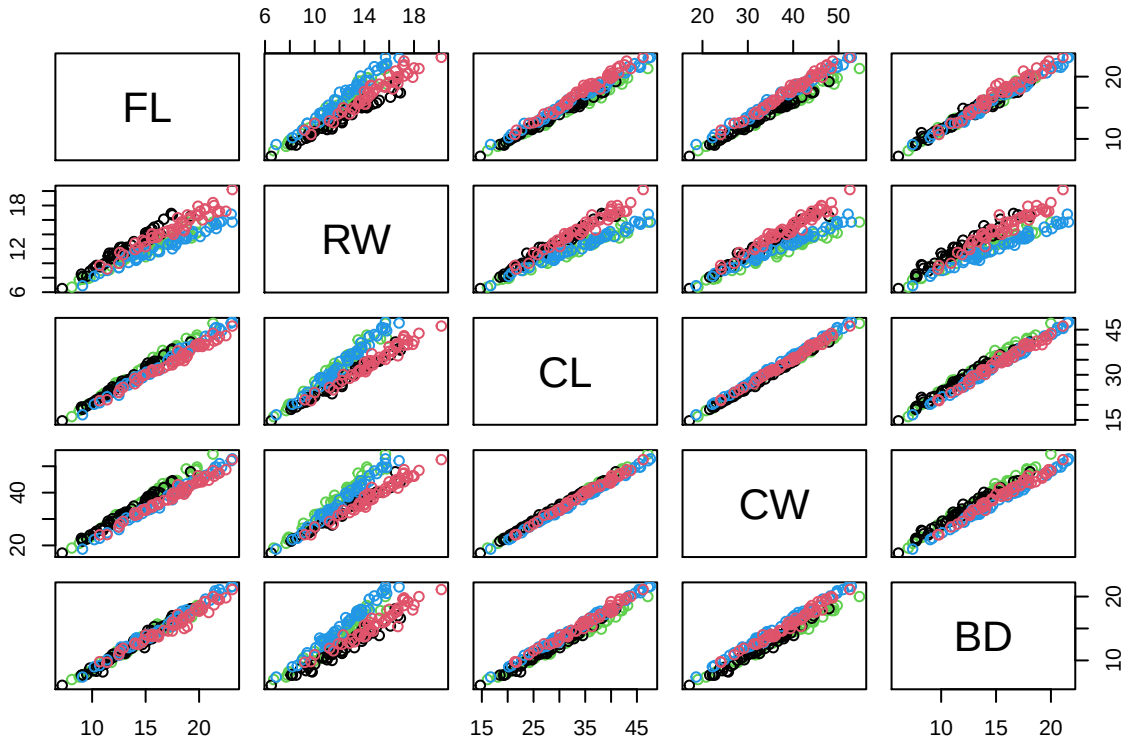

```
trueClasses <- paste(crabs$sex, crabs$sp, sep = "-")
table(trueClasses, KM$cluster)
```

```
##
## trueClasses  1  2  3  4
##         F-B 14 17 18  1
##         F-O 17  4 14 15
##         M-B 17 10 16  7
##         M-O 14  6 19 11
```

```
pairs(crabsquant, col = KM$cluster)
```




```
pairs(crabsquant, col = as.factor(trueClasses))
```



Clustering of raw data

```
TrueClasses_mat <- matrix(c(1,2,3,4), 2, 2)
TrueClasses_mat
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```
colnames(TrueClasses_mat) <- levels(crabs$sex)
rownames(TrueClasses_mat) <- levels(crabs$sp)
TrueClasses_mat
```

```
##   F M
## B 1 3
## O 2 4
```

```
TrueClasses <- diag(TrueClasses_mat[crabs$sex, crabs$sp])
TrueClasses
```

```
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4
## [112] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [149] 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [186] 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Truclass_mat associe a chaque couple (sp, sex) un numéro de classe. TrueClasses_mat[crabs$sex, crabs$sp] applique a chaque couple possible des 2 variables la valeur, seul la diagonale correspond au vrai couple.

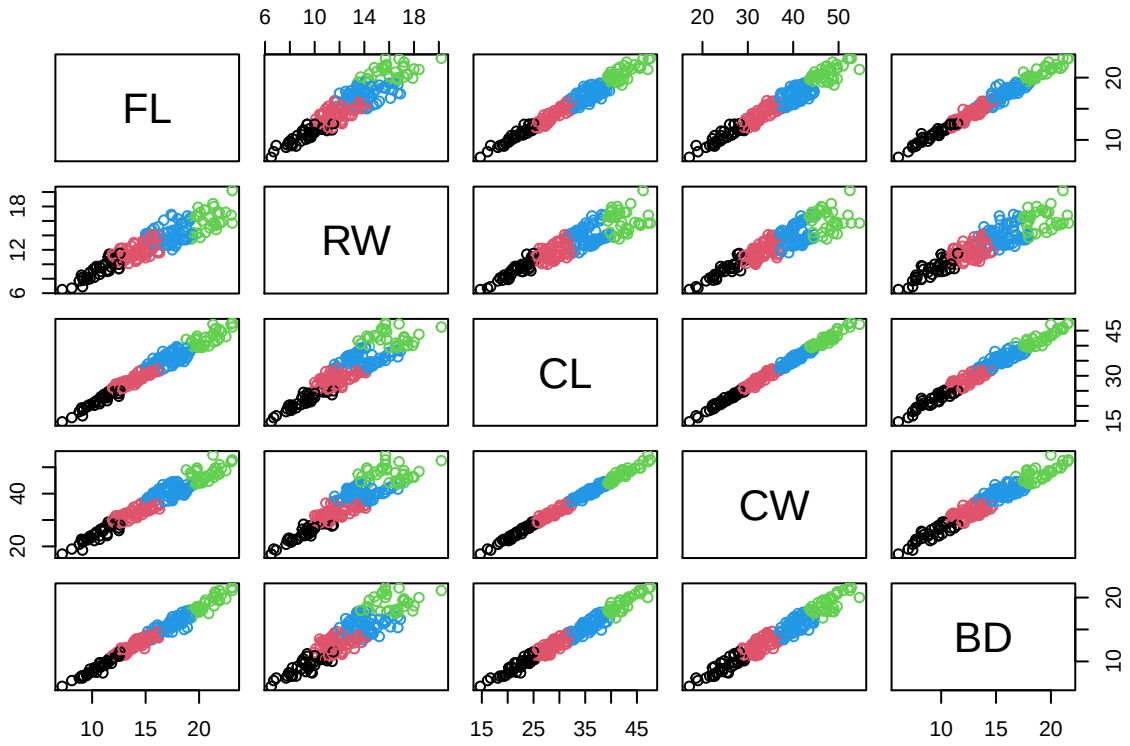
```
res <- kmeans(crabsquant, 4)
str(res)  # structure of an R project
```

```
## List of 9
##  $ cluster     : Named int [1:200] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:200] "1" "2" "3" "4" ...
##  $ centers     : num [1:4, 1:5] 10.6 14.16 20.79 17.24 9.16 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:5] "FL" "RW" "CL" "CW" ...
##  $ totss       : num 28500
##  $ withinss    : num [1:4] 798 836 631 776
##  $ tot.withinss: num 3041
##  $ betweenss   : num 25459
##  $ size        : int [1:4] 37 67 34 62
##  $ iter        : int 2
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

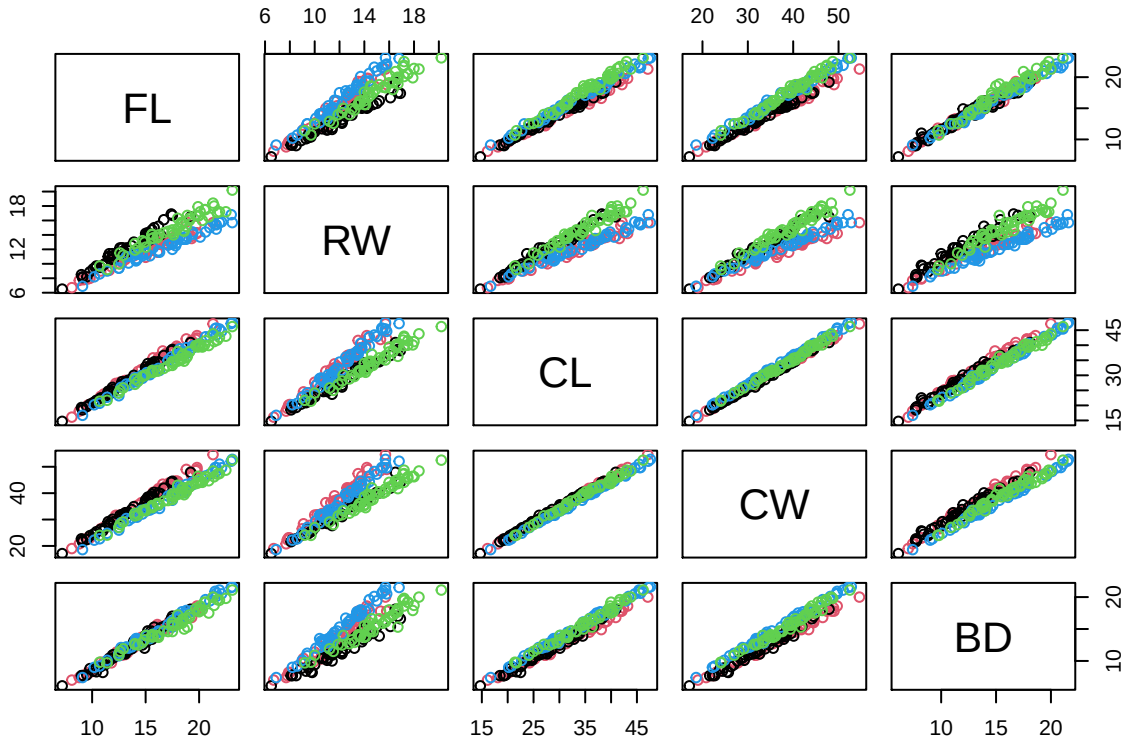
```
res
```

```
## K-means clustering with 4 clusters of sizes 37, 67, 34, 62
##
## Cluster means:
##         FL        RW       CL       CW        BD
## 1 10.59730  9.162162 21.62432 24.81081  9.124324
## 2 14.16269 11.843284 29.22687 33.17761 12.658209
## 3 20.79118 16.088235 42.48529 47.70882 19.097059
## 4 17.23710 14.003226 35.77903 40.64355 15.662903
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1   1   1   1   1   1   1   1   1   2   2   2   2   2   2   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   2   2   2   4   4   4   4   4   4   4   4   4   4   4   4   4   4
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   4   4   4   3   3   3   3   3   3   3   1   1   1   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   1   1   1   1   2   1   2   2   2   2   2   2   2   2   2   2   2   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   4   2   2   2   4   4   4   4   4   4   4   4   4   4   4   4   4   3
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   1   1   1   1   1   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   2   2   2   2   2   4   4   4   4   4   4   4   4   4   4   4   4   4   4   3
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   3   3   3   3   3   3   3   3   3   3   1   1   1   1   2   2   2   2   2   2
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   2   2   2   2   2   2   2   4   4   4   4   4   4   4   4   4   4   4   4
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   4   4   4   4   4   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##
## Within cluster sum of squares by cluster:
## [1] 797.8286 836.3722 630.8824 776.2439
##  (between_SS / total_SS =  89.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
pairs(crabsquant, col = res$cluster) # we have 4 classes, so 4 colors for clustering k-means
```

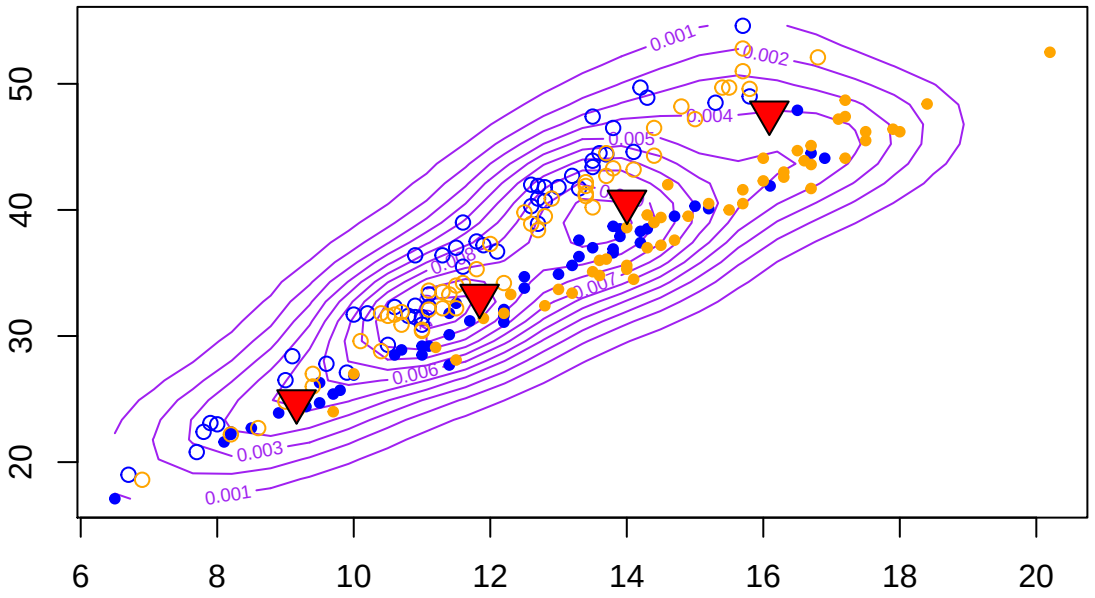



```
pairs(crabsquant, col = as.factor(TrueClasses))
```




```
z <- kde2d(crabsquant[,2], crabsquant[,4])
contour(z, col = "purple")
# placons les points dans le graphe de contour z
points(crabsquant[,c(2,4)], col = c("blue", "orange")[crabs$sp], pch = c(20,21)[crabs$sex])
```

```
# placons encore les centres des classes
points(res$center[,c(2,4)], cex = 2, pch = 25, bg = "red")
```



Clustering for raw data

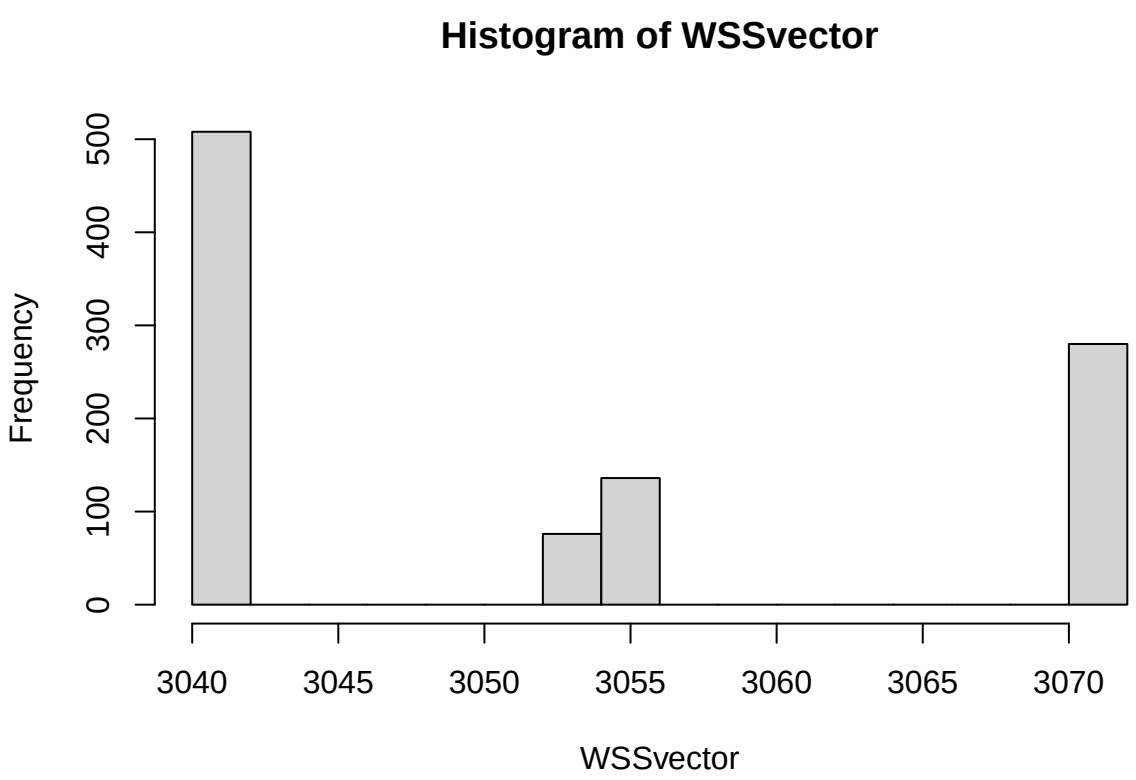
```
table(Kmeans = res$cluster, TrueClasses)
```

```
##       TrueClasses
## Kmeans  1  2  3  4
##      1 17 10  4  6
##      2 18 16 14 19
##      3  1  7 15 11
##      4 14 17 17 14
```

```
WSS <- function(partition) {
  sum(partition$withinss)
}
sortie = matrix(rep(0,4000), nrow = 1000, ncol = 4)
WSSvector <- rep(0,1000)
for (i in 1:1000) {
  res <- kmeans(crabsquant,4)
  sortie[i,] <- res$withinss
  WSSvector[i] <- WSS(res)
}
summary(WSSvector)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3041    3041    3041    3053    3072    3072
```

```
hist(WSSvector, breaks = 20)
```



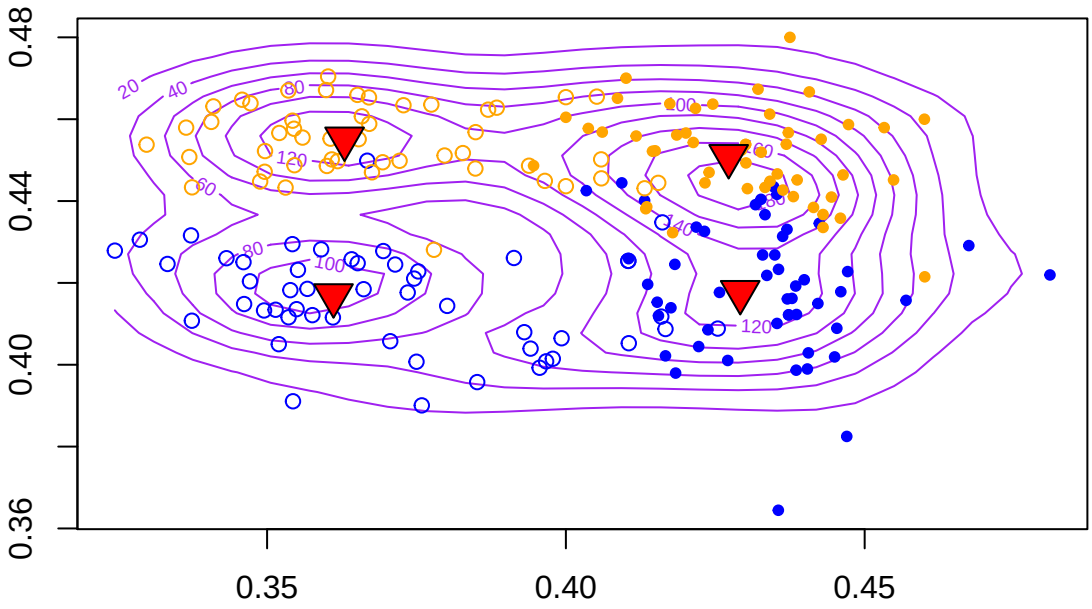
Tained a bad result because of size effect (latent factor): * the size information is present in all variables * this has a masking effect Solution: Divide all variables by one of them allows us to have features which are relative.

```
library(MASS)
n <- dim(crabs)[1]   # the number of rows, nb d'individus
dim(crabs)   # dimension of dataset "crabs
```

```
## [1] 200   8
```

```
crabsquant <- crabs[,4:8]
# faisons une simple petite transformation pour enlever l'effet de taille
crabsquant2 <- (crabsquant/crabsquant[,3])[,-3]
j=0
for (i in c(1,2,4,5)) {
  j = j + 1
  names(crabsquant2)[j] <- paste(names(crabsquant)[j],"/",names(crabsquant[3]))
}
```

```
res_bis <- kmeans(crabsquant2, 4)
z <- kde2d(crabsquant2[,2], crabsquant2[,4])
contour(z, col = "purple")
# placons les points dans le graphe de contour z
points(crabsquant2[,c(2,4)], col = c("blue", "orange")[crabs$sp], pch = c(20,21)[crabs$sex])
# placons encore les centres des classes
points(res_bis$center[,c(2,4)], cex = 2, pch = 25, bg = "red")
```



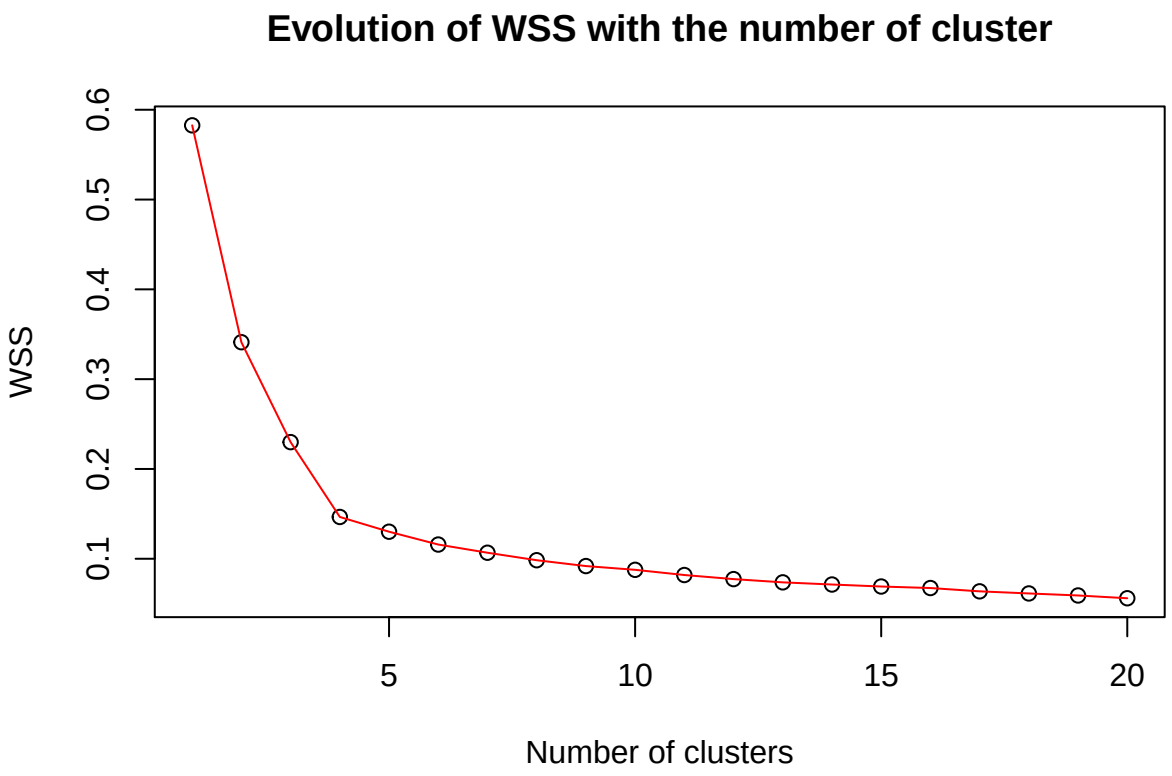
Clustering for the transformed dataset

```
table(Kmeans = res_bis$cluster, TrueClasses)
```

```
##       TrueClasses
## Kmeans  1  2  3  4
##      1  0 40  0  0
##      2 50 10  0  0
##      3  0  0  0 44
##      4  0  0 50  6
```

How many clusters ?

```
WSSkcluster <- rep(0,20)
for (k in 1:20) {
  WSSmax <- Inf
for (i in 1:10) {
  res <- kmeans(crabsquant2,k)
if (WSS(res) < WSSmax) {
  partition <- res
  WSSmax <- WSS(res)
}
}
  WSSkcluster[k] <- WSS(partition)
}
plot(WSSkcluster, xlab = "Number of clusters", ylab = "WSS", main = "Evolution of WSS with the number o
lines(WSSkcluster, col = "red")
```


Evolution of WSS with the number of cluster
WSS
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
20
Number of clusters