Recherche de partition

Christophe Ambroise

1 L'algorithme des centres mobiles

1.1 Historique

52

un algorithme réinventé dans de nombreux domaines 51

Lloyd (1957)

Edwards et Cavali Sforza (1965)

Macqueen (1967)

...

différent noms 51

k-means (C-moyenne)

centres mobiles,

réallocation centrage

algorithme LBG

1.2 Rappel des notations

52

Données : $\Omega = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^t$

Partition : $P = \{P_1, \ldots, P_K\}$

1.3 Description de l'algorithme

L'algorithme des centres-mobiles peut se définir ainsi :

1. Tirage au hasard de K points de Ω qui forment les centres initiaux des K classes.
2. Tant que non convergence
   (a) construction de la partition suivante en affectant chaque point de Ω à la classe dont il est le plus près du centre (en cas d'égalité, l'affectation se fait à la classe de plus petit indice).
   (b) les centres de gravité de la partition qui vient d'être calculée deviennent les nouveaux centres.

52

$L = (\lambda_1, \ldots, \lambda_K)$ représente un K-uple de $\mathbb{R}^p$

$P = (P_1, \ldots, P_K)$ une partition de Ω en K classes,

la suite construite par l'algorithme peut être notée sous la forme :

$$L^0 \to P^1 \to L^1 \to P^2 \to L^2 \to \ldots \to P^n \to L^n \to \ldots$$

1.4 Le critère

La qualité d'un couple partition-centres est mesurée par la somme des inerties des classes par rapport à leur centre :

$$C(P,L) = \sum_{k=1}^{K} I(P_k, \lambda_k) = \sum_{k=1}^{K} \sum_{x \in P_k} d^2(x, \lambda_k)$$

où

- $P = (P_1, P_2, \ldots, P_K)$,
- $L = (\lambda_1, \ldots, \lambda_K)$.

1.5 Convergence

On peut montrer qu'à chacune des deux étapes de l'algorithme, on améliore le critère C. Plus précisément, on a les relations suivantes :

$$C(P^{n+1}, L^n) \leq C(P^n, L^n) \tag{1}$$

$$C(P^{n+1}, L^{n+1}) \leq C(P^{n+1}, L^n) \tag{2}$$

1.6 Remarques

52

optimisation locale et non globale

en pratique convergence en moins de 10 itérations

1.7 Variantes

52

méthode séquentielle : kmeans de MacQueen

choix du nombre de classes : isodata

floue : fuzzy c-means

1.8 Analogie avec l'ACP

Comme L^q est fonction de P^q,

$$\begin{aligned} C(P^q, L^q) &= \sum_{k=1}^{K} I(P_k^q, \lambda_k^q) \\ &= \sum_{k} I(P_k^q) \\ &= I_W(P^q) \end{aligned}$$

Centres mobiles	ACP
Centres	Axes
Maximisation de l'inertie interclasse	Maximisation inertie projetée

2 Exemple d'application sous R

2.1 La fonction kmeans du module mva

```
K-Means Clustering

Description:

Perform k-means clustering on a data matrix.

Usage:

keans(x, centers, iter.max = 10)
```

```
Arguments:

x: A numeric matrix of data, or an object that can be coerced to
   such a matrix (such as a numeric vector or a data frame with
   all numeric columns).

 centers: Either the number of clusters or a set of initial cluster
   centers. If the first, a random set of rows in 'x' are chosen
   as the initial centers.

iter.max: The maximum number of iterations allowed.
```

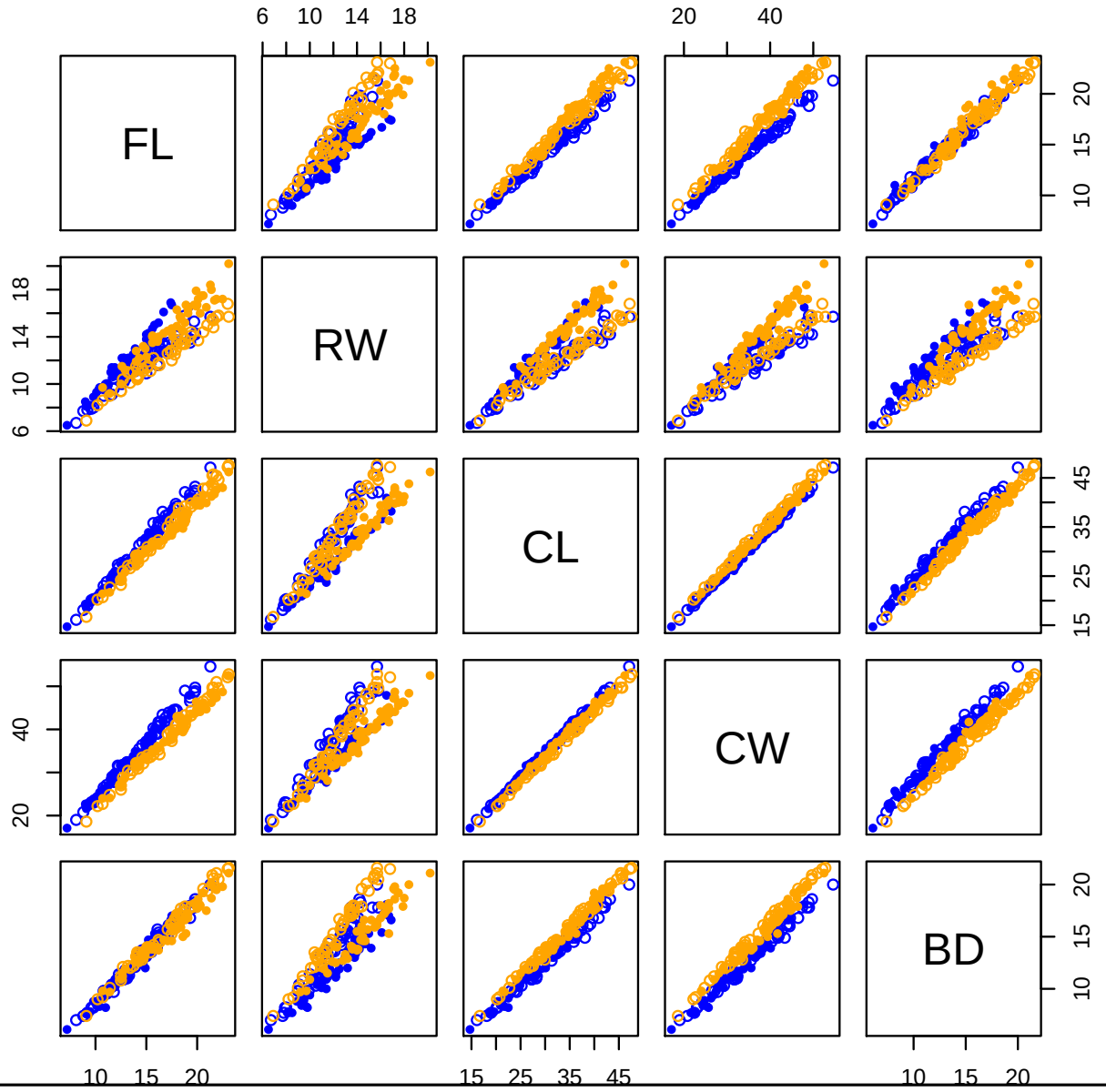
2.2 Data set : crabs (module MASS)

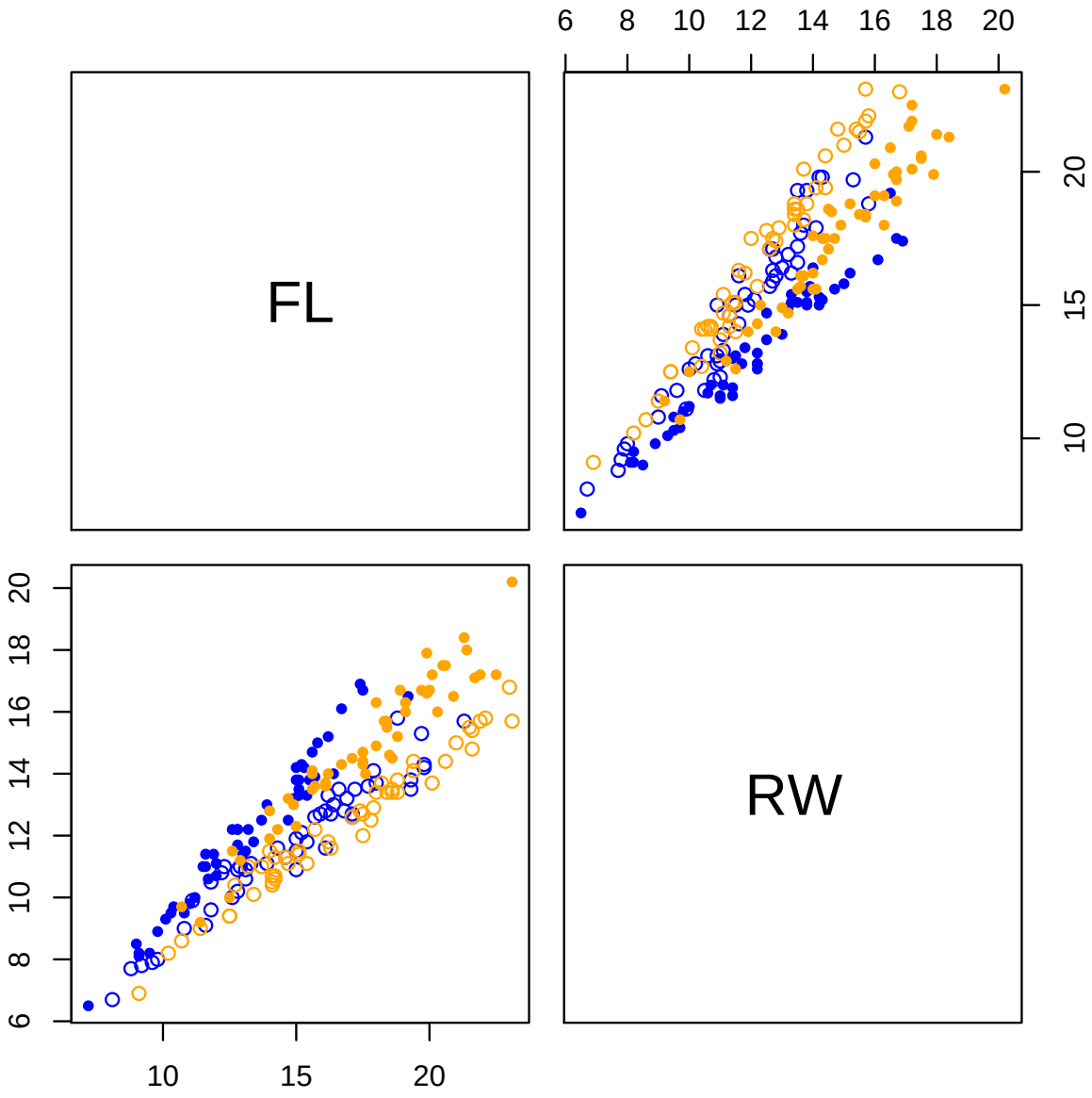
```
Morphological Measurements on Leptograpsus Crabs

Description:
The 'crabs' data frame has 200 rows and 8 columns, describing 5
morphological measurements on 50 crabs each of two colour forms
and both sexes, of the species _Leptograpsus variegatus_ collected
at Fremantle, W. Australia.

Usage:

data(crabs)
```

```
Format:
This data frame contains the following columns:
sp' 'species' - '"B"' or '"O"' for blue or orange
sex' as it says
index' index 1:50 within each of the four groups
FL' frontal lobe size (mm)
RW' rear width (mm)
CL' carapace length (mm)
CW' carapace width (mm)
BD' body depth (mm)

Source:
   Campbell, N.A. and Mahon, R.J. (1974) A multivariate study of
   variation in two species of rock crab of genus _Leptograpsus._
   _Australian Journal of  Zoology_ *22*, 417-425.
```

2.3 Graphes bivariables

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


FL
RW
CL
CW
BD
6 10 14 18
20 40
10 15 20
6 10 14 18
15 25 35 45
20 40
10 15 20
15 25 35 45
10 15 20


6
8
10
12
14
16
18
20
FL
RW
10
15
20

2.4 Classification des données brutes (1)

```
> res <- kmeans(crabsquant, 4)
> str(res)

List of 4
 $ cluster : int [1:200] 1 1 1 1 1 1 1 1 1 1 ...
 $ centers : num [1:4, 1:5] 10.60 20.79 14.16 17.24  9.16 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:4] "1" "2" "3" "4"
  .. ..$ : chr [1:5] "FL" "RW" "CL" "CW" ...
 $ withinss: num [1:4] 798 631 836 776
 $ size    : int [1:4] 37 34 67 62
```

2.5 Classification des données brutes (2)

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

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

55 Pourquoi ?

2.6 L'effet taille

52

l'information taille est présente dans toutes les variables

effet de masque

55 Diviser toutes les variables par une seule permet de limiter l'effet taille

2.7 Variabilité des résultats

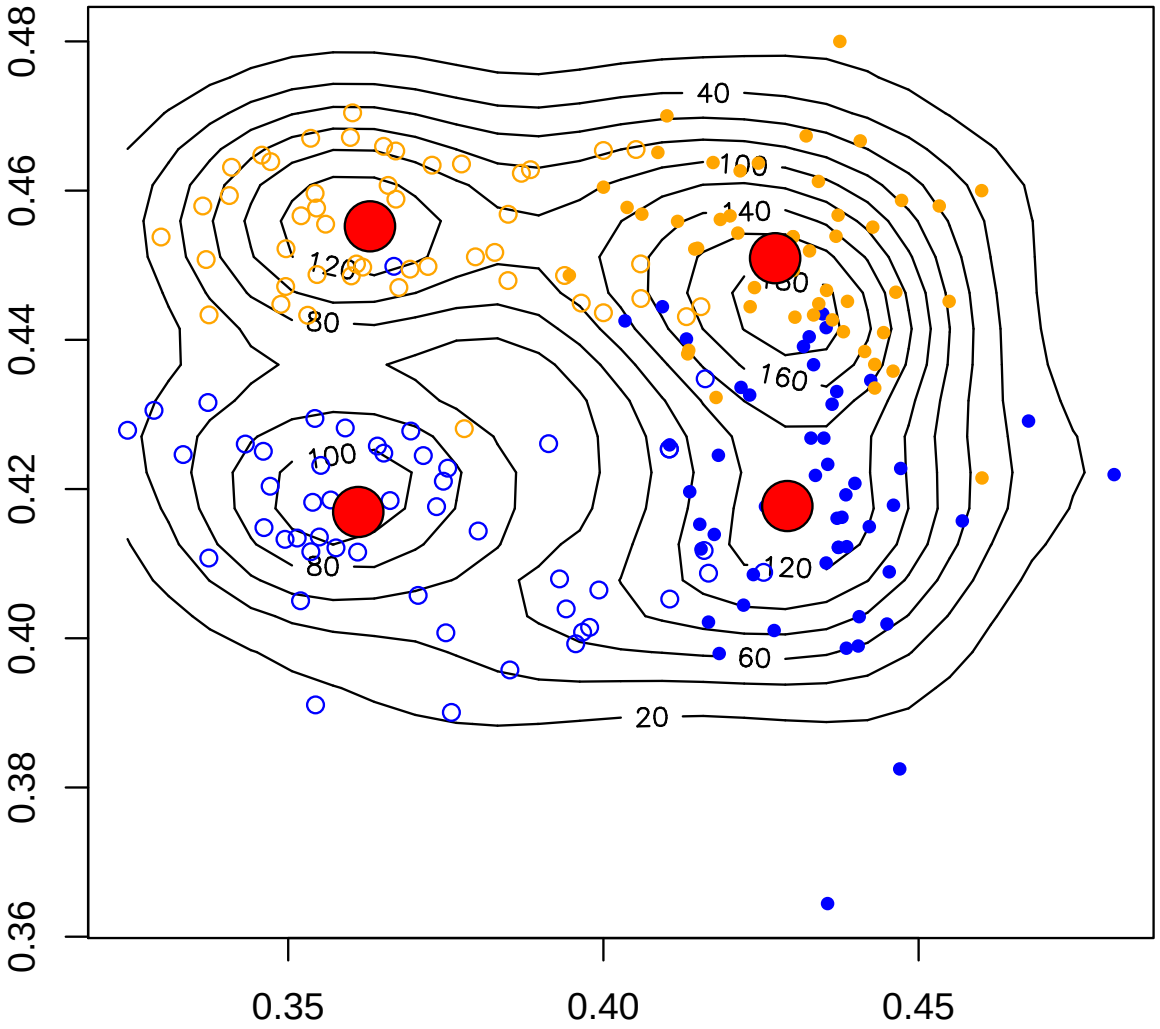
```
> WSS <- function(partition) {
+     sum(partition$withinss)
+ }
> WSSvector <- rep(0, 100)
> for (i in 1:100) {
+     res <- kmeans(crabsquant2, 4)
+     WSSvector[i] <- WSS(res)
+ }
> summary(WSSvector)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1465  0.1465  0.1465  0.1465  0.1465  0.1465
```

2.8 Supression de l'effet taille

```
> data(crabs)
> n = dim(crabs)[1]
> crabsquant <- crabs[, 4:8]
> n = dim(crabs)[1]
> crabsquant <- crabs[, 4:8]
> crabsquant2 <- (crabsquant/crabsquant[, 3])[,
+     -3]
> j = 0
> for (i in c(1, 2, 4, 5)) {
+     j = j + 1
+     names(crabsquant2)[j] <- c(paste(names(crabsquant)[i],
+         "/", names(crabsquant[3])))
+ }
> res <- kmeans(crabsquant2, 4)
```

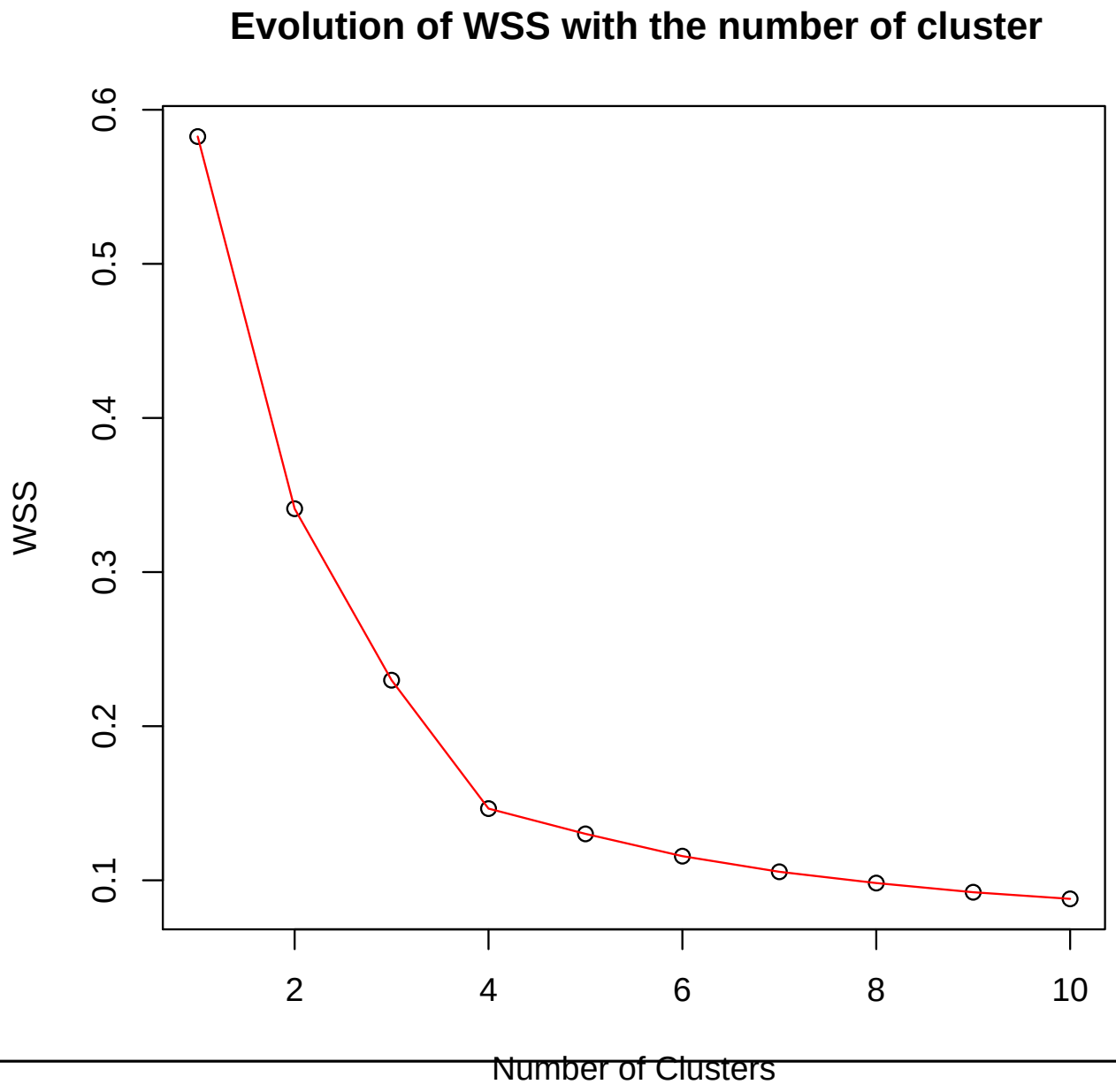
```
> z <- kde2d(crabsquant2[, 2], crabsquant2[,
+     4])
> contour(z)
> points(crabsquant2[, c(2, 4)], col = c("blue",
+     "orange")[crabs$sp], pch = c(20, 21)[crabs$sex])
> points(res$center[, c(2, 4)], cex = 3, , pch = 21,
+     bg = "red")
```


0.48
0.46
0.44
0.42
0.40
0.38
0.36
0.35
0.40
0.45
40
100
140
120
80
160
100
80
120
60
20

2.9 Combien de classes

```
> WSS1cluster <- sum(diag((n - 1) * var(crabsquant2)))
> WSSkcluster <- rep(0, 10)
> WSSkcluster[1] <- WSS1cluster
> for (k in 2:10) {
+     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 of clus
> 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
2
4
6
8
10
Number of Clusters