TP4: Classification hiérarchique

DJEBALI Wissam

3 mars 2018

Classification hiérarchique

packges R : factoextra, cluster, NbClust

L'algorithme de segmentation hiérarchique ascendante est disponible au travers de la fonction hclust() de R.

Elle s'applique non pas à un jeu de données, mais à une matrice de distance. On peut facilement obtenir cette matrice pour un data frame à l'aide de la fonction dist() qui calcule la distance euclidienne entre chaque paire de données du data frame.

Classification hiérarchique ascendante ou Agglomerative clustering=AGNES (Agglomerative Nesting)

Principe : Chaque individu est initialement considéré comme un groupe (feuille de l'arbre). À chaque étape de l'algorithme, les 2 groupes les plus similaires sont combinés dans un nouveau groupe(noeud de l'arbre). Cette procédure est répétée jusqu'à ce que tous les individus fasse partie du même groupe. Opposé à la méthode de **Classification hiérarchique descendante ou Divisive clustering =DIANA (Divise Analysis)**, qui elle part d'un groupe et fait l'inverse.

AGNES *est une bonne méthode pour identifier un petit nombre de groupes.*

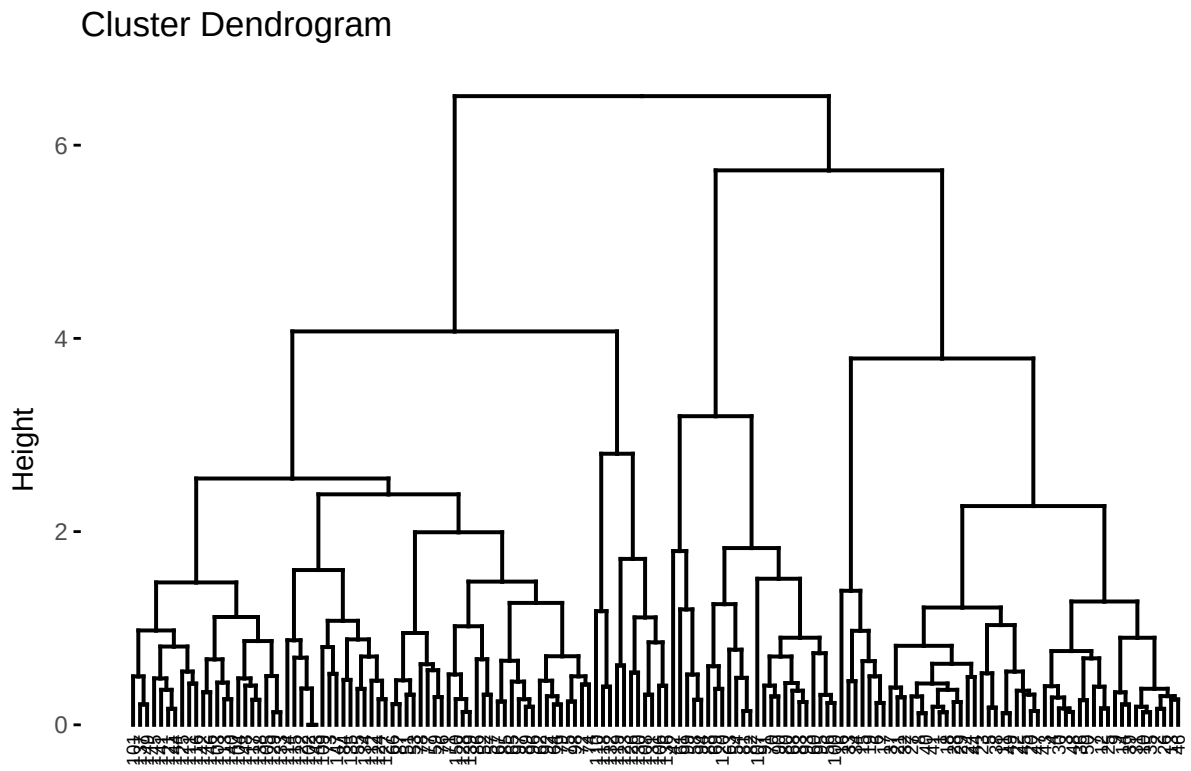
```
# Préparation des données
ir <-scale(iris[, -5])

head(ir)
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]   -0.8976739  1.01560199    -1.335752   -1.311052
## [2,]   -1.1392005 -0.13153881    -1.335752   -1.311052
## [3,]   -1.3807271  0.32731751    -1.392399   -1.311052
## [4,]   -1.5014904  0.09788935    -1.279104   -1.311052
## [5,]   -1.0184372  1.24503015    -1.335752   -1.311052
## [6,]   -0.5353840  1.93331463    -1.165809   -1.048667
```

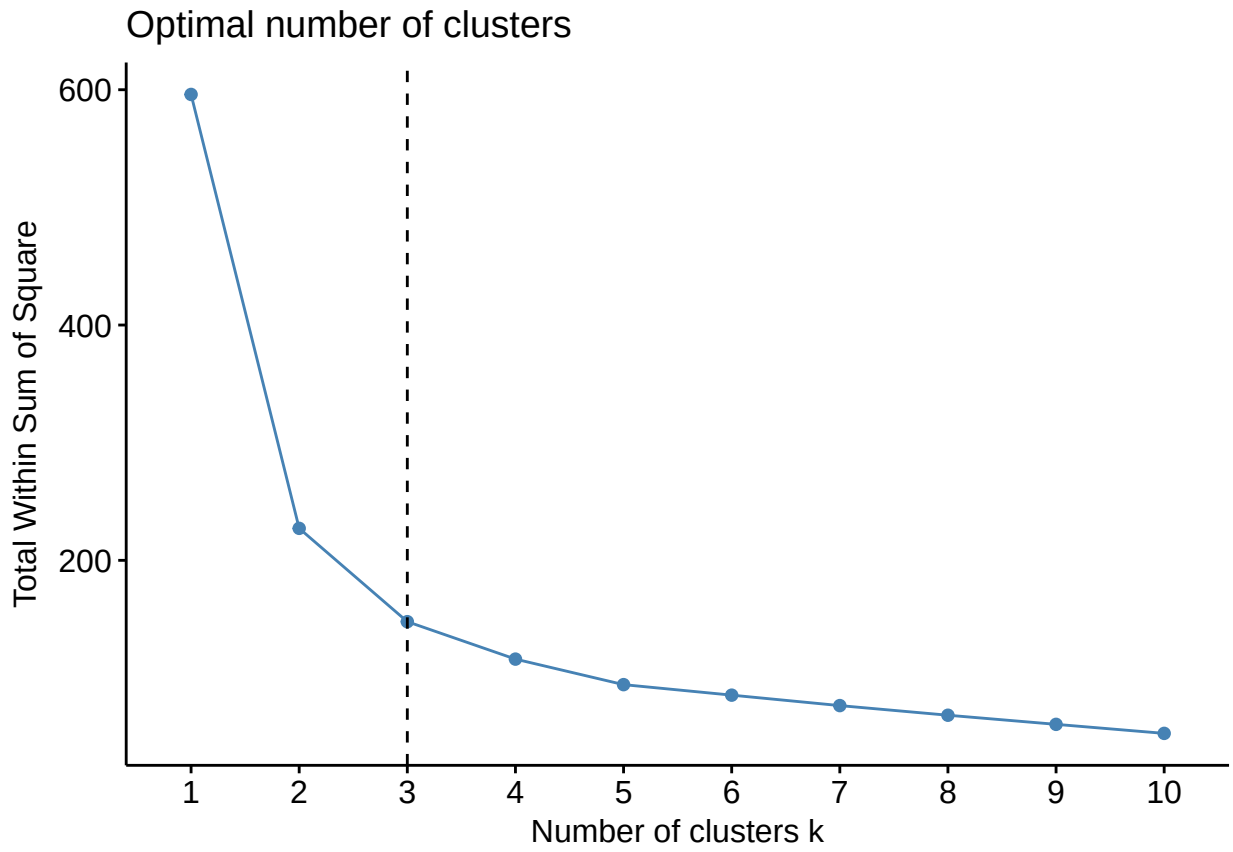
```
# Pour décider de la similarité entre deux groupes, on utilise différentes distances

# Calcul de la distance euclidienne entre chaque paire
res.dist <- dist(ir, method = "euclidean")

# Affichage des distances entre les individus
as.matrix(res.dist)[1:6, 1:6]
```

```
##           1         2         3         4         5         6
## 1 0.0000000 1.1722914 0.8427840 1.0999999 0.2592702 1.0349769
## 2 1.1722914 0.0000000 0.5216255 0.4325508 1.3818560 2.1739229
## 3 0.8427840 0.5216255 0.0000000 0.2829432 0.9882608 1.8477070
## 4 1.0999999 0.4325508 0.2829432 0.0000000 1.2459861 2.0937597
## 5 0.2592702 1.3818560 0.9882608 1.2459861 0.0000000 0.8971079
## 6 1.0349769 2.1739229 1.8477070 2.0937597 0.8971079 0.0000000
```

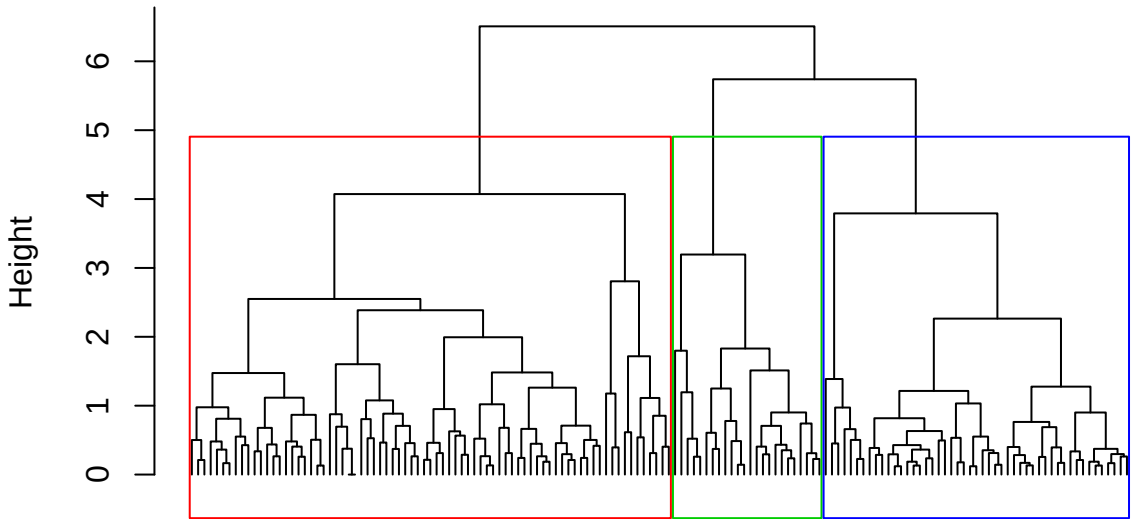
```r
# Résultats de la classification hiérarchique
hc <- hclust(res.dist, method = "complete")

# Visualisation de hclust
#plot(hc, labels = FALSE, hang = -1)
fviz_dend(hc, cex = 0.5)
```




```r
# Détermination du nombre de groupes : Méthode de Elbow
fviz_nbclust(ir, hcut, method = "wss")+ geom_vline(xintercept = 3, linetype = 2)
```




```
# On peut voir que le meilleur choix du nb de grp est 3

# Visualisation de hclust
plot(hc, labels = FALSE, hang = -1)
# Add rectangle around 3 groups
rect.hclust(hc, k = 3, border = 2:4)
```




```
# Calcul de la distance cophentic
res.coph <- cophenetic(hc)
# Correlation entre la distance cophenetic et distance euclidienne
cor(res.dist, res.coph)
```

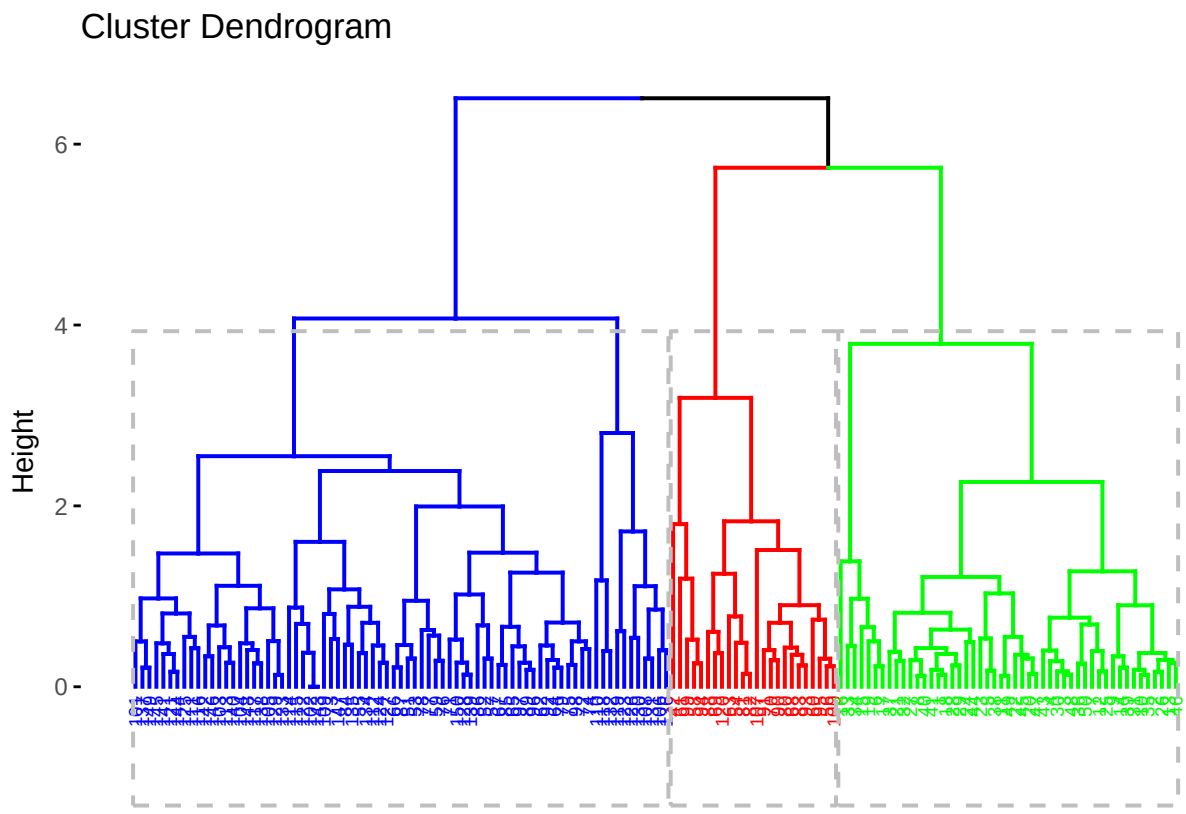
```
## [1] 0.7514592
```

```
# Plus la corrélation est proche de 1 plus le choix découpage en groupe des individus est précis
# À partir de 0.75 on juge en général qu'un découpage est précis

# Découpe de l'arbre en 3 groupes
grp<-cutree(hc, 3)

# Tableau d'effectif des groupes
table(grp,iris[,5])
```

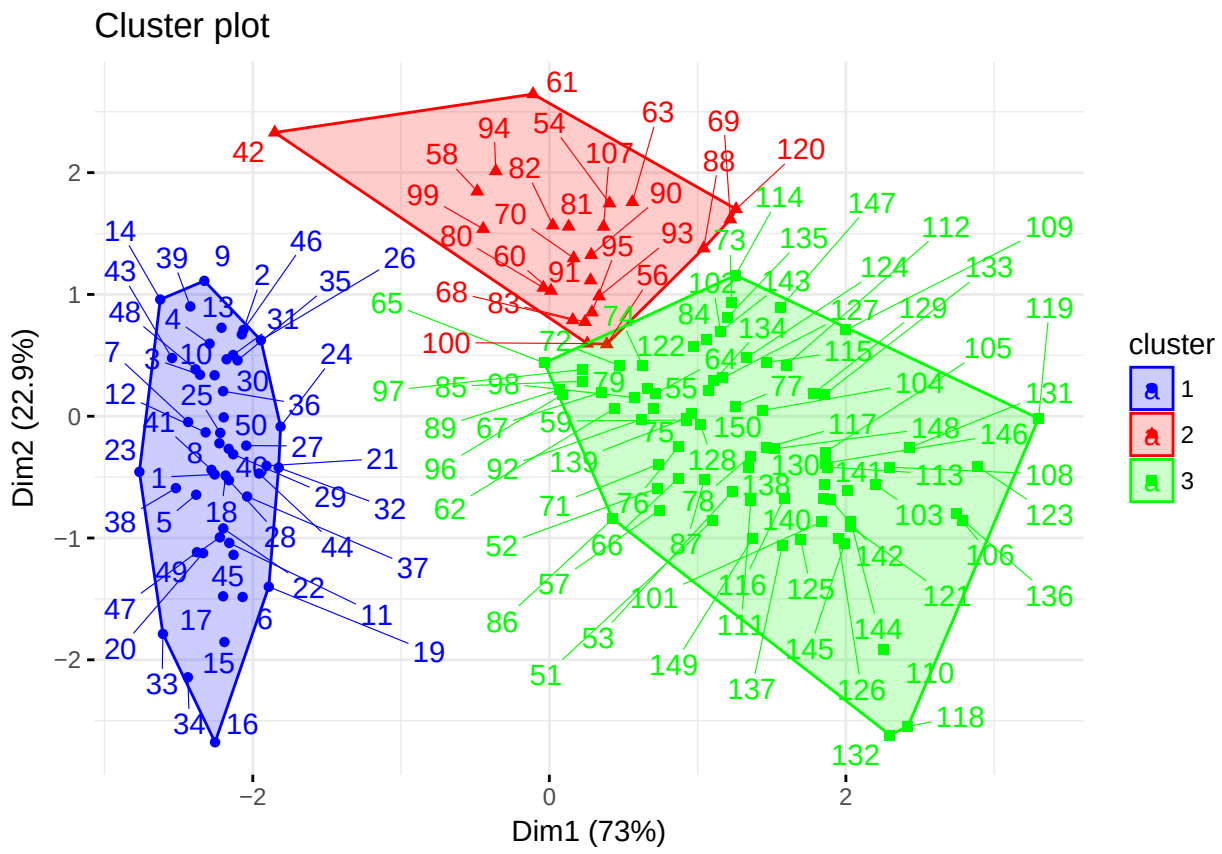
```
##
## grp setosa versicolor virginica
##   1     49          0         0
##   2      1         21         2
##   3      0         29        48
```

```
# Découpage en 3 groupes et coloration par groupes
fviz_dend(hc, k = 3, # Cut in four groups
          cex = 0.5, # label size
          k_colors = c("blue", "red", "green"),
          color_labels_by_k = TRUE, # color labels by groups
          rect = TRUE # Add rectangle around groups
```

```
)
```

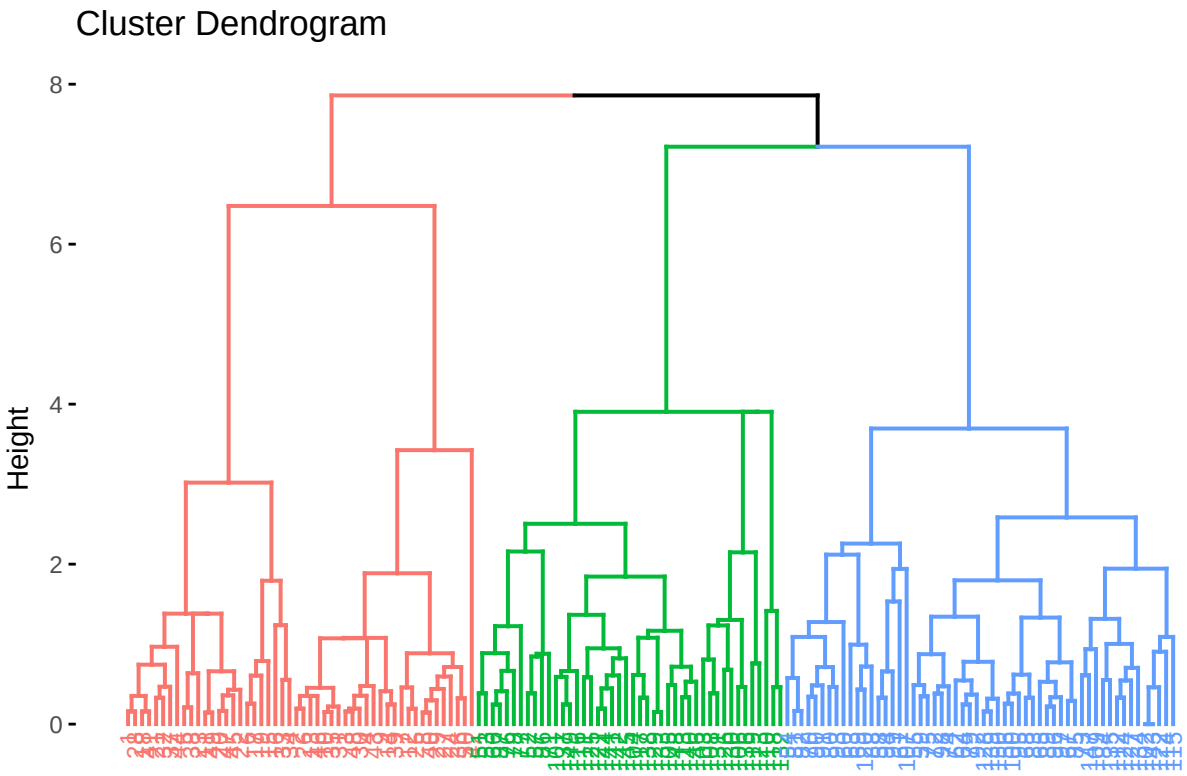



```
# Visualisation des individus en fonction de leur groupe
fviz_cluster(list(data = ir, cluster = grp),
             palette = c("blue", "red", "green"),
             ellipse.type = "convex", # Concentration ellipse
             repel = TRUE, # Avoid label overplotting (slow)
             show.clust.cent = FALSE, ggtheme = theme_minimal())
```



Classification hiérarchique descendante ou Divisive clustering=DIANA (Divise Analysis)

```
library("cluster")
res.diana <- diana(x = ir, # data matrix
                   stand = TRUE, # standardize the data
                   metric = "euclidean" # metric for distance matrix
                   )

# Visualisation de l'arbre ou dendogramme
fviz_dend(res.diana, cex = 0.6, k = 3)
```


Cluster Dendrogram
Height
8
6
4
2
0