Projet 5/5 Clustering

Alexandre CORRIOU - Nicolas SALVAN

2024-05-23

Ce document a pour objectif de réaliser une analyse de clustering, il contient toutes les sorties R et les commentaires associés.

Lecture des données nettoyées

Importation du dataset

```
data <- read.csv("data/data_cleaned.csv", header = TRUE, sep = ",")
data$diagnosis <- as.factor(data$diagnosis)
```

Aperçu rapide

```
head(data)
```

```
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 1         M       17.99        10.38         122.80    1001.0         0.11840
## 2         M       20.57        17.77         132.90    1326.0         0.08474
## 3         M       19.69        21.25         130.00    1203.0         0.10960
## 4         M       11.42        20.38          77.58     386.1         0.14250
## 5         M       20.29        14.34         135.10    1297.0         0.10030
## 6         M       12.45        15.70          82.57     477.1         0.12780
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## 1          0.27760         0.3001             0.14710        0.2419
## 2          0.07864         0.0869             0.07017        0.1812
## 3          0.15990         0.1974             0.12790        0.2069
## 4          0.28390         0.2414             0.10520        0.2597
## 5          0.13280         0.1980             0.10430        0.1809
## 6          0.17000         0.1578             0.08089        0.2087
##   fractal_dimension_mean radius_se texture_se perimeter_se area_se
## 1                0.07871    1.0950     0.9053        8.589  153.40
## 2                0.05667    0.5435     0.7339        3.398   74.08
## 3                0.05999    0.7456     0.7869        4.585   94.03
## 4                0.09744    0.4956     1.1560        3.445   27.23
## 5                0.05883    0.7572     0.7813        5.438   94.44
## 6                0.07613    0.3345     0.8902        2.217   27.19
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
```

```
## 1      0.006399        0.04904      0.05373           0.01587     0.03003
## 2      0.005225        0.01308      0.01860           0.01340     0.01389
## 3      0.006150        0.04006      0.03832           0.02058     0.02250
## 4      0.009110        0.07458      0.05661           0.01867     0.05963
## 5      0.011490        0.02461      0.05688           0.01885     0.01756
## 6      0.007510        0.03345      0.03672           0.01137     0.02165
##   fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## 1             0.006193        25.38         17.33          184.60     2019.0
## 2             0.003532        24.99         23.41          158.80     1956.0
## 3             0.004571        23.57         25.53          152.50     1709.0
## 4             0.009208        14.91         26.50           98.87      567.7
## 5             0.005115        22.54         16.67          152.20     1575.0
## 6             0.005082        15.47         23.75          103.40      741.6
##   smoothness_worst compactness_worst concavity_worst concave.points_worst
## 1           0.1622            0.6656          0.7119               0.2654
## 2           0.1238            0.1866          0.2416               0.1860
## 3           0.1444            0.4245          0.4504               0.2430
## 4           0.2098            0.8663          0.6869               0.2575
## 5           0.1374            0.2050          0.4000               0.1625
## 6           0.1791            0.5249          0.5355               0.1741
##   symmetry_worst fractal_dimension_worst
## 1         0.4601                 0.11890
## 2         0.2750                 0.08902
## 3         0.3613                 0.08758
## 4         0.6638                 0.17300
## 5         0.2364                 0.07678
## 6         0.3985                 0.12440
```

```
# str(data)
# summary(data)
```

Séparation des données numériques et centrage-réduction

```
data_num <- data[-1]
data.cent <- scale(data_num)
```

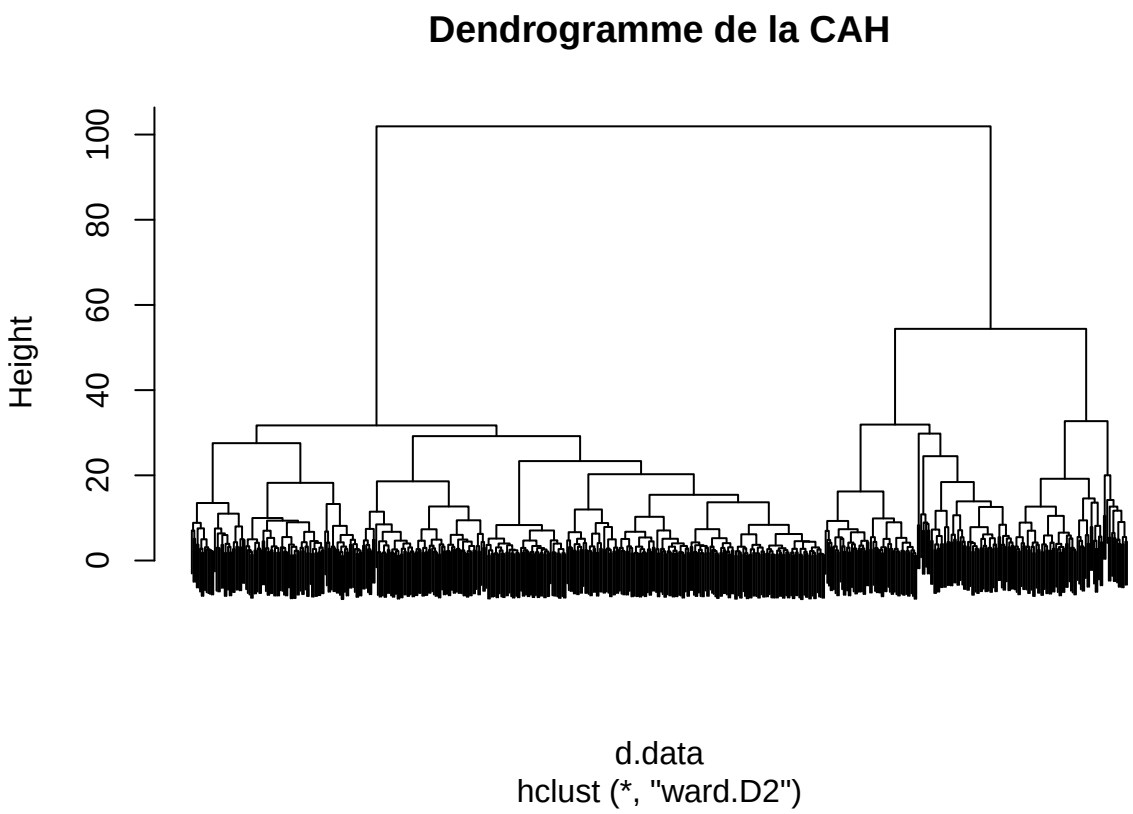
Modèles de clustering

Comme nous avons relativement peu d'individus, nous allons réaliser dans un premier temps une classification ascendante hiérarchique (CAH) pour avoir une idée du nombre de clusters à choisir, puis nous réaliserons un k-means pour obtenir les clusters.

CAH - Classification Ascendante Hiérarchique*

Dendrogramme

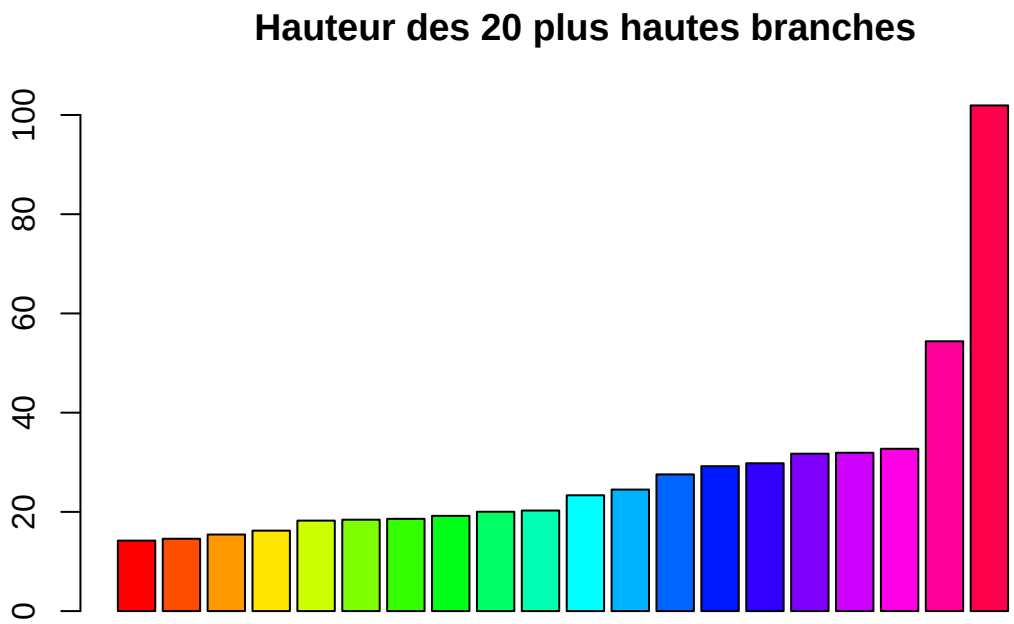
```
d.data = dist(data.cent)
hc <- hclust(d.data, method = "ward.D2")
plot(hc, cex = 0.01, main = "Dendrogramme de la CAH" )
```



On peut voir qu'il y a 3 branches qui sont assez longues, on peut donc choisir 3 clusters. On peut le confirmer en affichant la hauteur des branches les plus hautes.

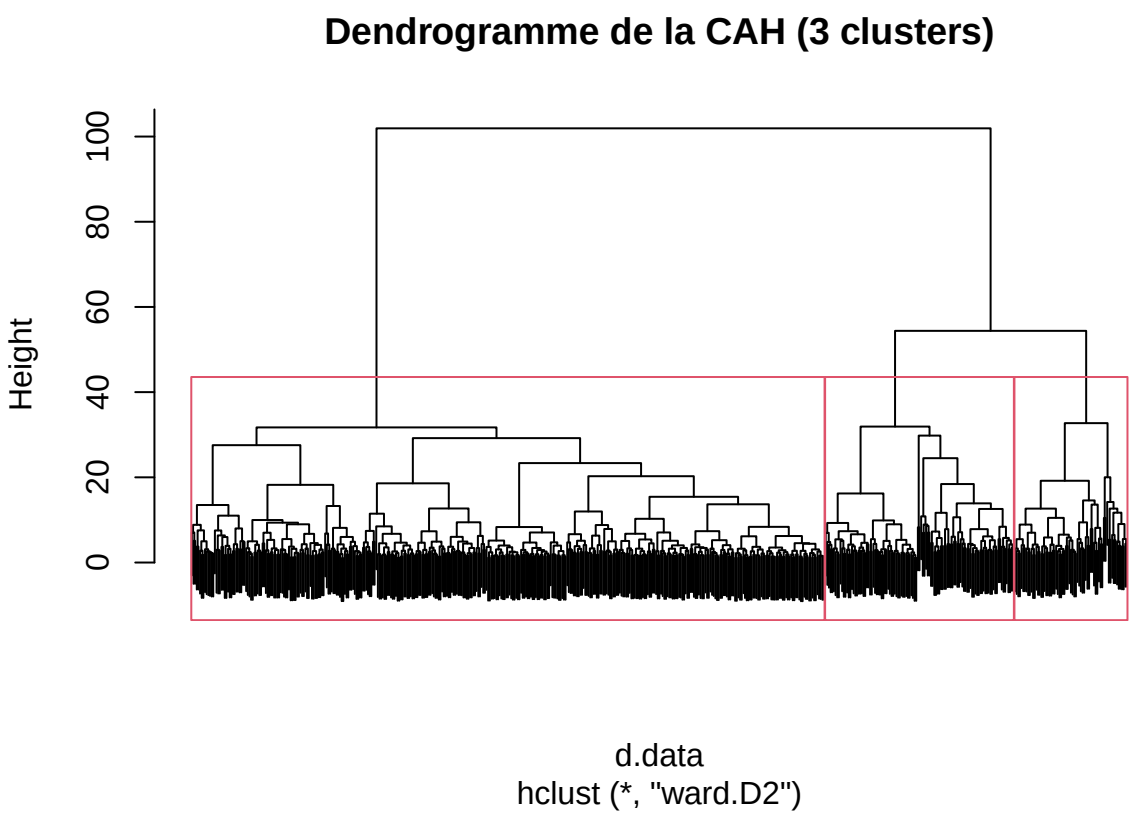
Hauteur des branches les plus hautes

```
N <- 20
barplot(hc$height[(dim(data)[1]-N):dim(data)[1]], col = rainbow(N), main = "Hauteur des 20 plus hautes
```



On peut bien déceler 3 branches qui sont assez longues. Nous choisirons donc 3 clusters pour la suite de l'analyse.

```
plot(hc, cex = 0.01, main = "Dendrogramme de la CAH (3 clusters)")
rect.hclust(hc, k = 3)
```



K-means

Clustering à 3 groupes

On calcule le clustering à 3 groupes en utilisant 1000 points de départ différents pour éviter les minima locaux.

```
kmeans.result=kmeans(data.cent, nstart = 1000, centers=3)
```

On constitue une base de données avec les données centrées-réduites et la classe de chaque individu obtenue avec le clustering.

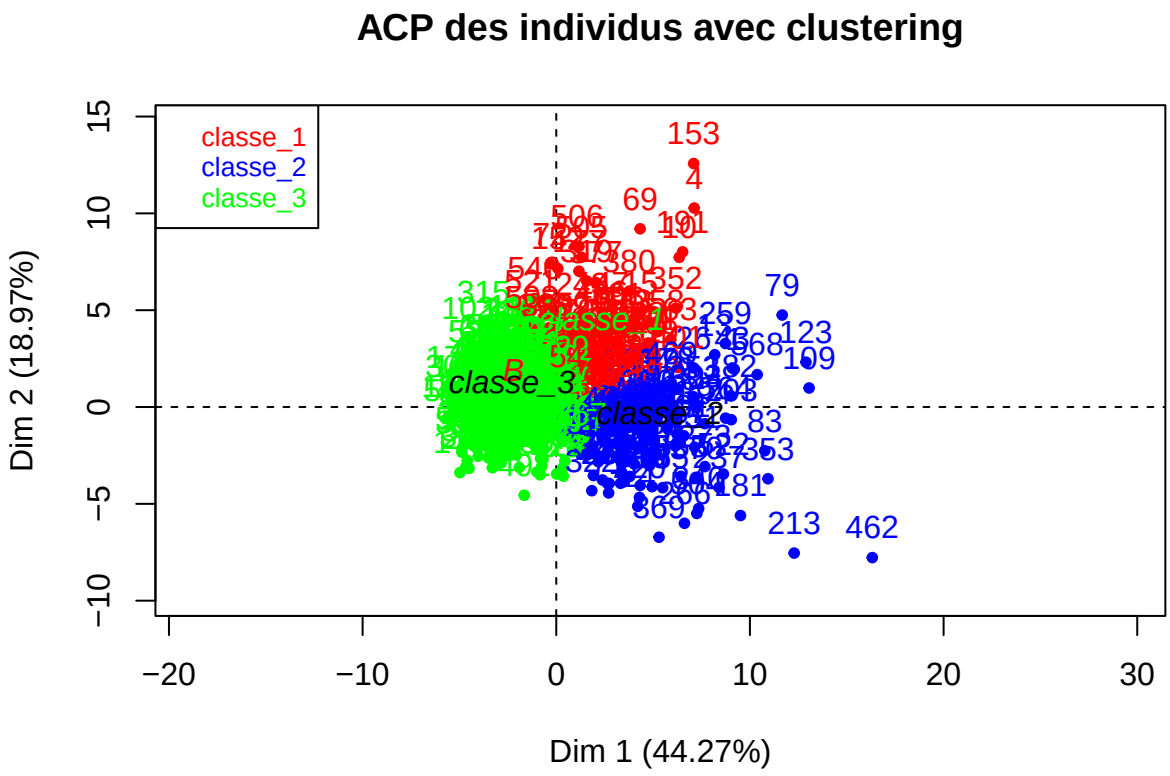
```
data_classe <- cbind.data.frame(data[1], data.cent, classe=factor(kmeans.result$cluster))
```

Visualisation des clusters avec une ACP

```
library(FactoMineR)
```

```
res.pca <- PCA(data_classe, graph = FALSE, quali.sup = c(1, 32))
```

```r
plot(res.pca, choix = "ind", graph.type = "classic", habillage = 32, col.hab = c("red", "blue", "green"))
```



On observe bien sur le plan principal de l'ACP que les clusters sont bien séparés.

Comparaison des clusters trouvés avec le diagnostic

On peut comparer les clusters trouvés avec le diagnostic initial pour voir si les clusters correspondent bien aux diagnostics.

```r
table(data_classe$diagnosis, data_classe$classe)
```

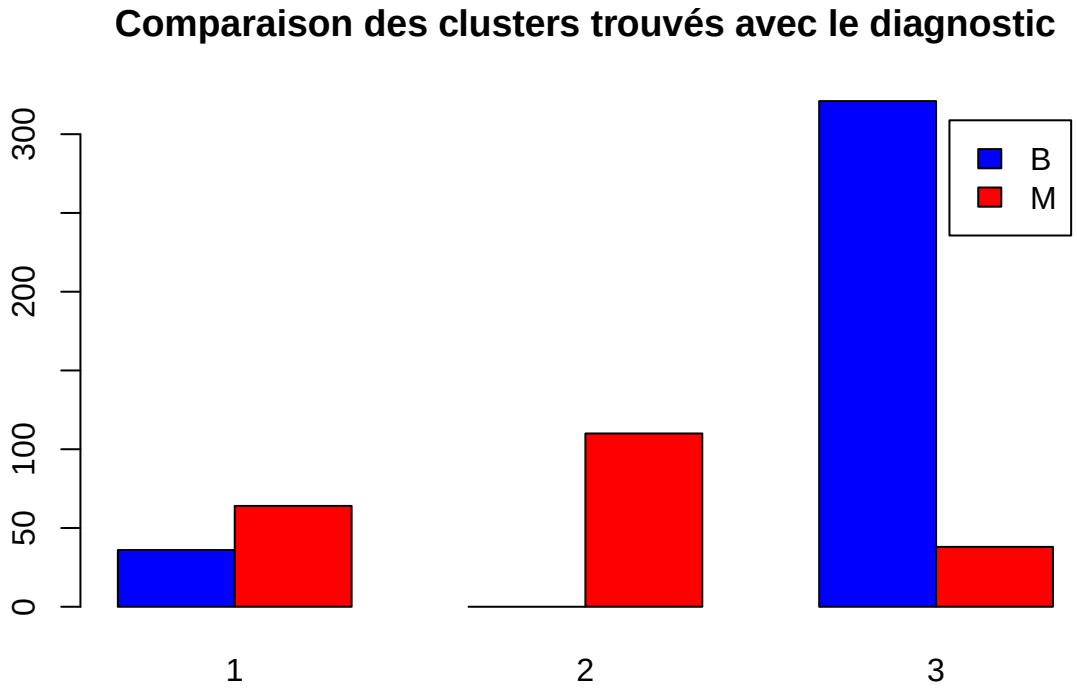
```
##
##       1   2   3
##   B  36   0 321
##   M  64 110  38
```

```r
table(data_classe$classe, data_classe$diagnosis)
```

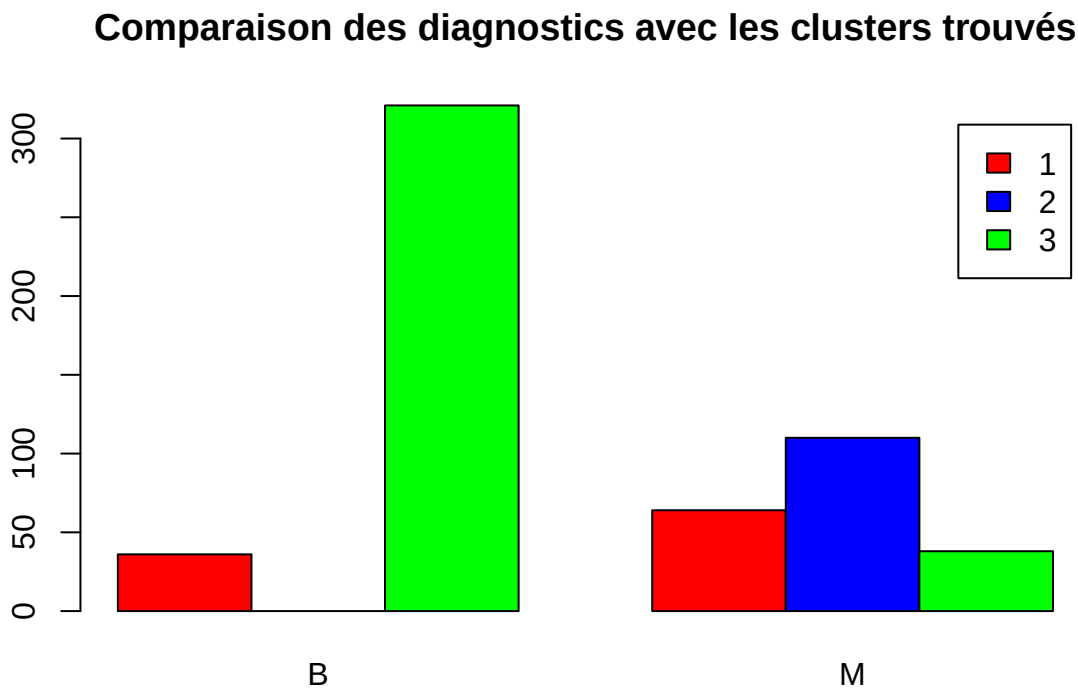
```
##
##       B   M
##   1  36  64
##   2   0 110
##   3 321  38
```

On observe que les clusters ne correspondent pas aux diagnostics. En effet, les clusters 1 et 2 correspondent à des diagnostics plutôt malins, et le cluster 3 correspond à des diagnostics plutôt bénins. Seul le cluster 2 correspond bien aux diagnostics malins.

```
barplot(table(data_classe$diagnosis, data_classe$classe), beside = TRUE, col = c("blue", "red"), legend
```




```
barplot(table(data_classe$classe, data_classe$diagnosis), beside = TRUE, col = c("red", "blue", "green"
```

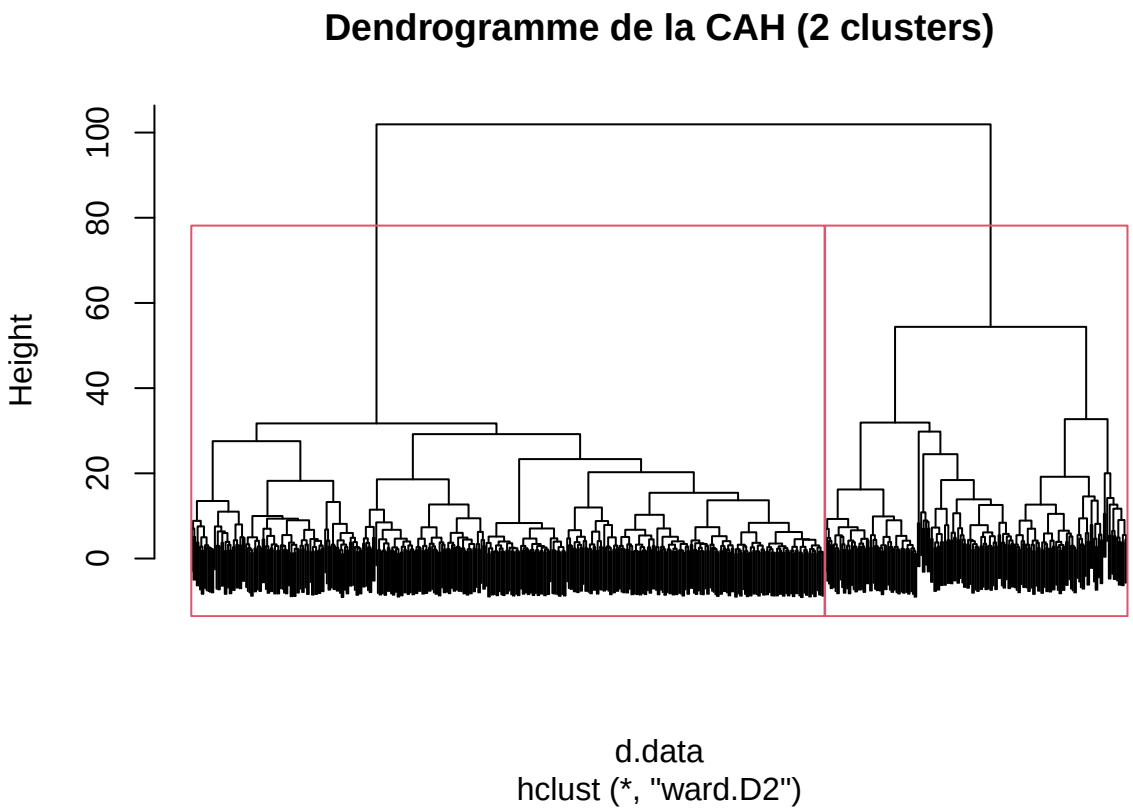


On peut visualiser ces résultats dans les histogrammes précédents.

Clustering à 2 groupes

On peut également réaliser un clustering à 2 groupes pour voir si les clusters correspondent mieux aux diagnostics.

```
plot(hc, cex = 0.01, main = "Dendrogramme de la CAH (2 clusters)")
rect.hclust(hc, k = 2)
```

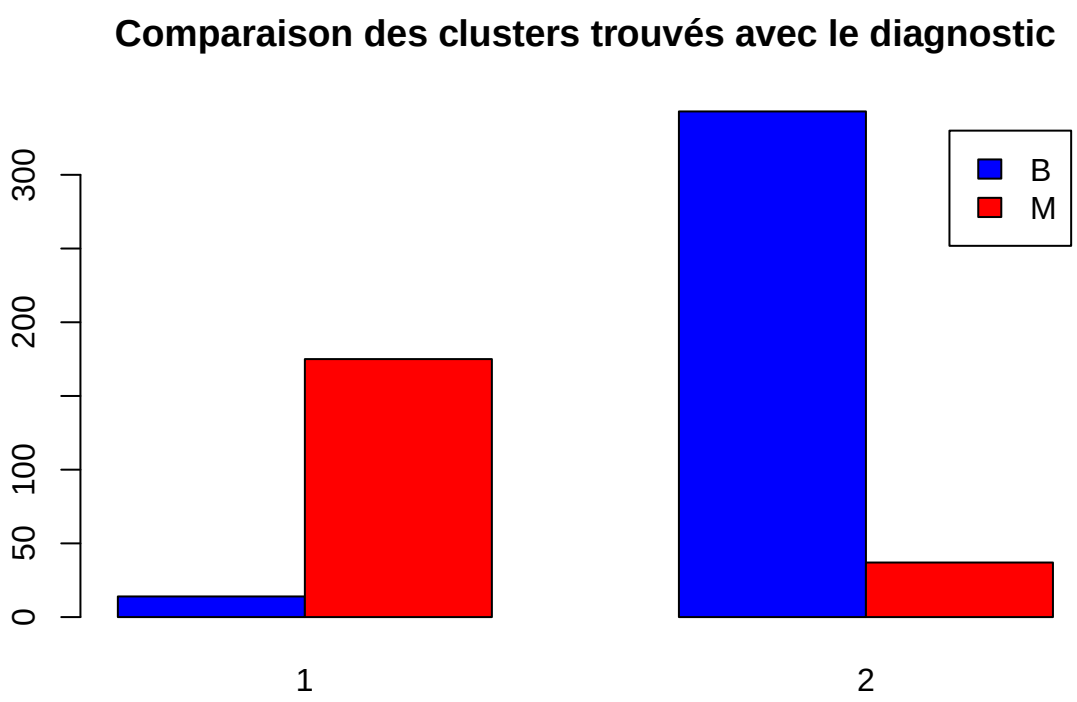



```
kmeans.result2=kmeans(data.cent, nstart = 1000, centers=2)
data_classe2 <- cbind.data.frame(data[1], data.cent, classe=factor(kmeans.result2$cluster))
table(data_classe2$diagnosis, data_classe2$classe)
```

```
##
##       1   2
##   B  14 343
##   M 175  37
```

On observe que les clusters correspondent mieux aux diagnostics. En effet, le cluster 1 correspond aux diagnostics bénins, et le cluster 2 correspond aux diagnostics malins.

```
barplot(table(data_classe2$diagnosis, data_classe2$classe), beside = TRUE, col = c("blue", "red"), lege
```



La classification à 2 groupes semble donc plus pertinente pour une analyse de clustering.

Conclusion

Nous avons réalisé une analyse de clustering sur les données nettoyées. Nous avons choisi de réaliser une CAH pour déterminer le nombre de clusters à choisir, puis un k-means pour obtenir les clusters. Nous avons choisi 3 clusters. Nous avons ensuite réalisé une ACP pour visualiser les clusters. Nous avons comparé les clusters trouvés avec les diagnostics initiaux. Nous avons observé que les clusters ne correspondaient pas forcément aux diagnostics. En effet, les clusters 1 et 2 correspondent à des diagnostics plutôt malins, et le cluster 3 correspond à des diagnostics plutôt bénins. Seul le cluster 2 correspond bien aux diagnostics malins.

Il est ainsi possible de déceler plusieurs catégories de tumeurs, mais il est plus difficile de savoir s'il s'agit de tumeurs bénignes ou malignes, d'où l'intérêt de réaliser une classification supervisée (ou pour les docteurs de faire d'autres tests).