TP/DM2 : classification non supervisée

Nous nous intéressons à des données de criminalité aux Etats-Unis. Le jeu de données comprends une ligne par Etat et 4 colonnes contenant les variables suivantes :

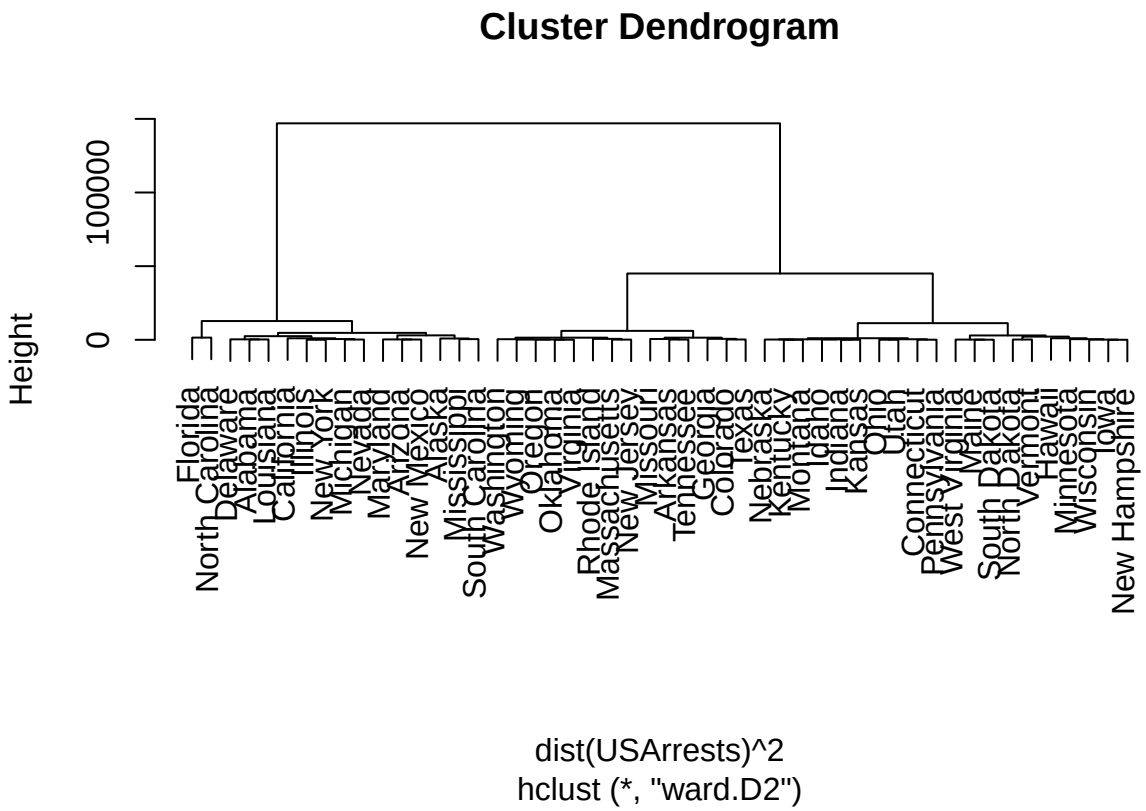
- `Murder` : nombre d'arrestations pour meurtre pour 100 000 habitants;
- `Assault` : nombre d'arrestations pour agression pour 100 000 habitants;
- `UrbanPop` : pourcentage de population urbaine;
- `Rape` : nombre d'arrestations pour viol pour 100 000 habitants.

L'objectif est de déterminer des profils type d'états en fonction de ces 4 variables.

Classification ascendante hiérarchique (CAH)

Nous allons réaliser un dendrogramme des Etats à l'aide de la méthode de Ward.

```
res.hclust = hclust(dist(USArrests)^2,method="ward.D2")
plot(res.hclust)
```



Question 1: Commenter le dendrogramme et proposer un choix du nombre de classes.

La méthode de ressemblance utilisée est celle du critère de Ward. Elle fonctionne de la manière suivante :

- On initialise en prenant un inidividu par classe

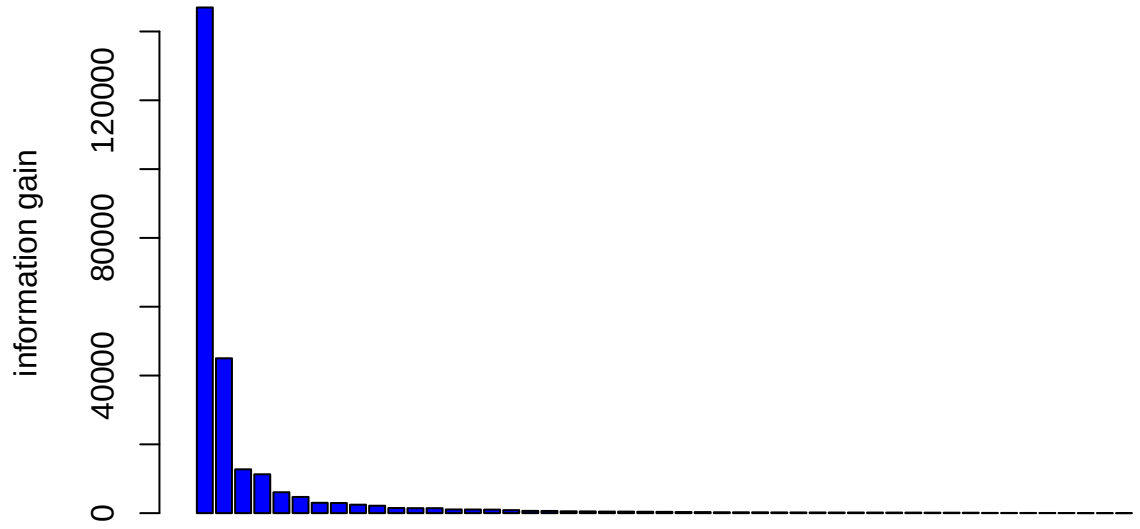
- On agrege à chaque étape les classes qui minimisent la diminution de l'intertie inter-classes

Pour rappel, l'inertie inter-classes est la somme des carrés des distances de chaque centre de gravité de classes au centre de gravité global du tableau. Comme vu dans le cours, on chercher donc à minimiser :

$$\frac{m_a m_b}{m_a + m_b} d^2(a,b) \quad \text{où :} \quad \begin{array}{l} - \ m_a \text{ et } m_b \text{ nombre d'individu dans les classes } a \text{ et } b \\ - \ d^2(a,b) \text{ la distance entre les barycentres des classes } a \text{ et } b \end{array}$$

On va alors regarder l'histogramme du gain d'intertie lors de l'augmentation du nombre de classes :

```
barplot(sort(res.hclust$height, decreasing = TRUE), col = 'blue', ylab = "information gain", xlab = "fr
```



On constate que les écarts sont importants lors du passage de une à deux classes, de deux à trois classes et de trois à quatre classes. On va calculer rapidement dans ces cas l'information qu'on récupère :

```
inertia_gain <- sort(res.hclust$height, decreasing = TRUE)
cat("information recuperee lors du passage de 1 a 2 classes : " ,(sum(inertia_gain[1])/sum(inertia_gain
```

```
## information recuperee lors du passage de 1 a 2 classes :  58.03606  %
```

```
cat("information recuperee lors du passage de 2 a 3 classes : " ,(sum(inertia_gain[2])/sum(inertia_gain
```

```
## information recuperee lors du passage de 2 a 3 classes :  17.77211  %
```

```
cat("information recuperee lors du passage de 3 a 4 classes : " ,(sum(inertia_gain[3])/sum(inertia_gain
```

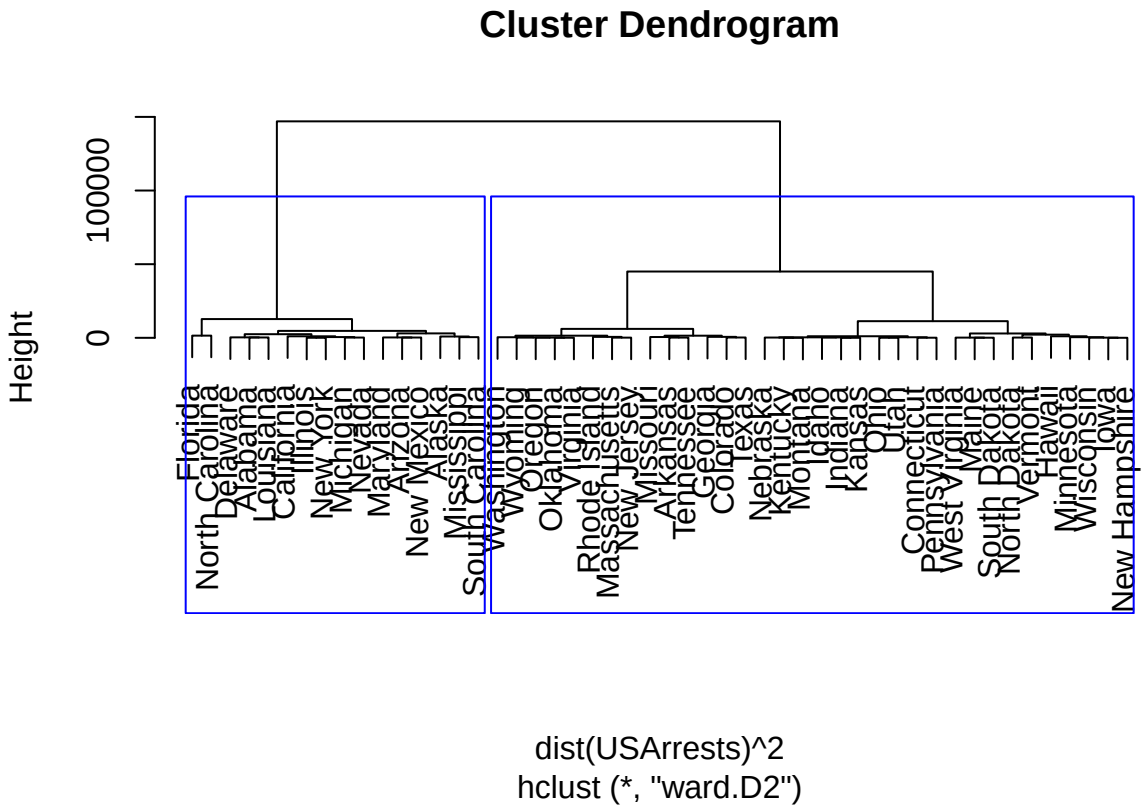
```
## information recuperee lors du passage de 3 a 4 classes :  5.027966  %
```

```
cat("information recuperee lors du passage de 4 a 5 classes : " ,(sum(inertia_gain[4])/sum(inertia_gain
```

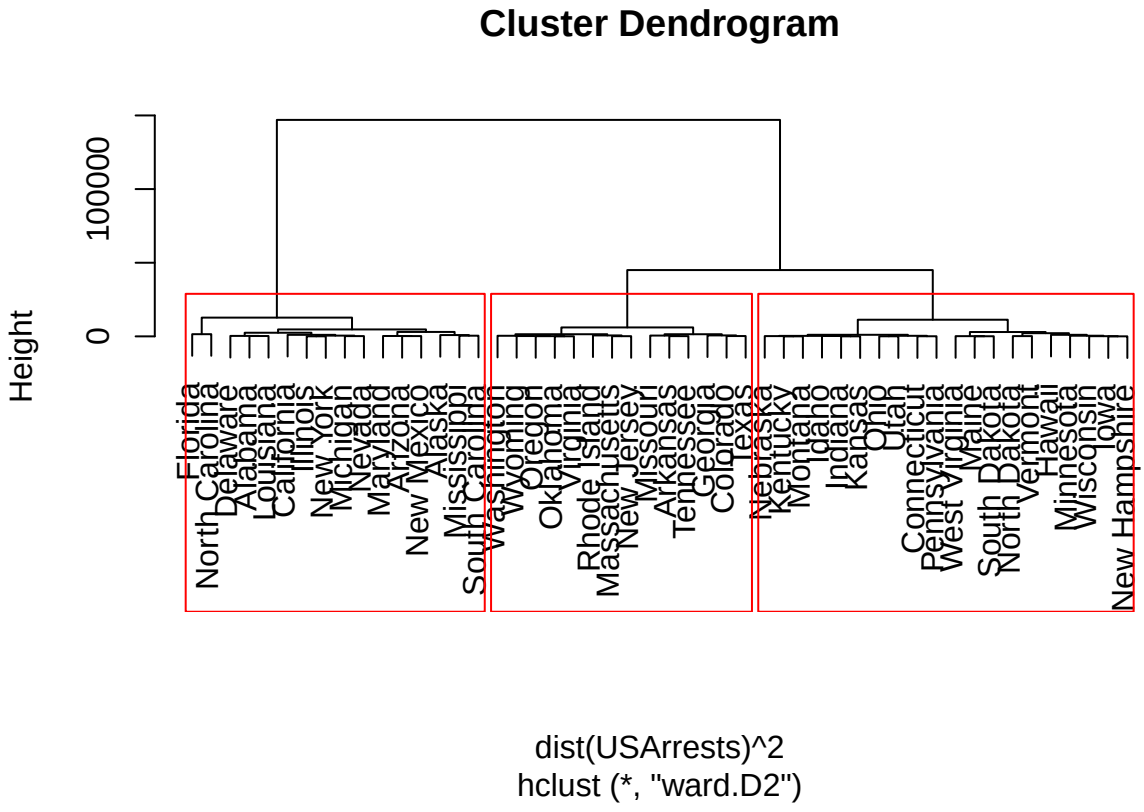
```
## information recuperee lors du passage de 4 a 5 classes :  4.457878  %
```

On peut donc décider selectionner 2 ou 3 classes pour récupérer 58% ou 76% de la variablité de l'information :

```
plot(res.hclust)
rect.hclust(res.hclust, 2, border = "blue")
```




```
plot(res.hclust)
rect.hclust(res.hclust, 3, border = "red")
```



Etant donné les arbres trouvés, celle à 3 classes semble donc la plus pertinente.

Méthode des k-moyennes

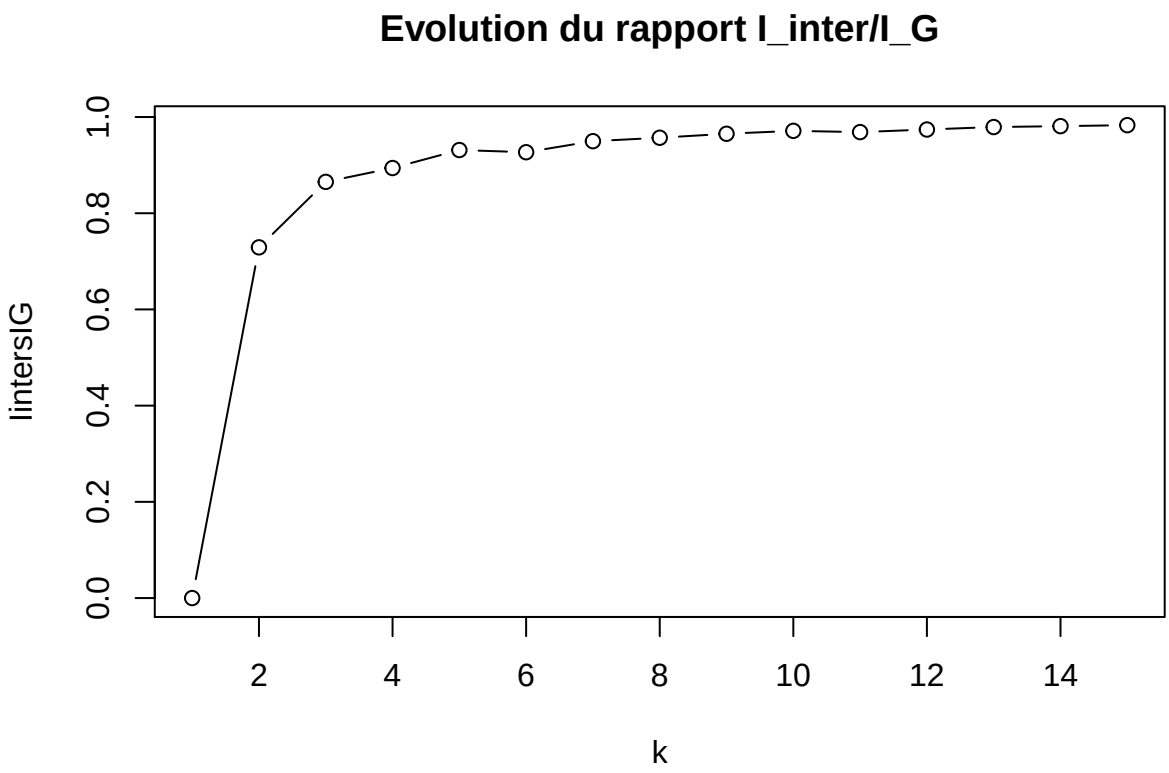
Nous utiliserons dans un premier temps la fonction `kmeans()`. Consulter l'aide de la fonction `kmeans()` et lancer la fonction pour un nombre de classes k que vous choisirez. Vous pouvez faire plusieurs essais avec la même valeur de k ou des valeurs différentes.

Le résultat de la fonction `kmeans()` vous donne :

- les moyennes des variables de chaque cluster,
- la classification des individus (ici les individus sont les Etats),
- la somme des carrés des distances à l'intérieur de chaque classe (`Within cluster sum of squares by cluster`) qui mesure l'homogénéité de chaque classe,
- le rapport $\mathcal{I}_{inter}/\mathcal{I}_G$ (noté `between_SS/total_SS`).

Pour choisir k, nous allons tracer l'évolution du rapport $\mathcal{I}_{inter}/\mathcal{I}_G$ en fonction de k.

```
kvect = 1:15
IintersIG = rep(NA,length(kvect))
for (j in 1:length(kvect)){
  res = kmeans(USArrests,kvect[j])
  IintersIG[j] = res$betweenss/res$totss
}
plot(kvect,IintersIG,type='b',main='Evolution du rapport I_inter/I_G',xlab='k')
```



Nous voyons que le rapport $\mathcal{I}_{inter}/\mathcal{I}_G$ augmente beaucoup lorsque l'on passe de 1 à 2 classes ou, dans une moindre mesure, de 2 à 3 classes. Au-delà de 3 classes, le rapport $\mathcal{I}_{inter}/\mathcal{I}_G$ croît peu. Cela nous oriente vers une classification à 2 ou 3 classes.

Une fois choisi le nombre de classes, nous allons déterminer un profil type par classe.

```r
k=3
res3 = kmeans(USArrests,k)
rbind(res3$centers,colMeans(USArrests))
```

```
##      Murder  Assault UrbanPop     Rape
## 1  4.270000  87.5500 59.75000 14.39000
## 2  8.214286 173.2857 70.64286 22.84286
## 3 11.812500 272.5625 68.31250 28.37500
##    7.788000 170.7600 65.54000 21.23200
```

```r
res3$cluster
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              3              3              3              2              3
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              1              3              3              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              1              1              3              1              1
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              1              1              3              1              3
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              2              3              1              3              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              1              1              3              1              2
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              3              3              3              1              1
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              2              1              2              3
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              1              2              2              1              1
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              2              1              1              2
```

Nous obtenons une classe avec des Etats ayant une criminalité basse (comprenant les Etats du Connecticut, Hawaï, Idaho,... attention cela peut varier pour différentes exécutions des k-moyennes à cause de l'aléa du à l'initialisation), une classe avec des Etats ayant une criminalité moyenne (Arkansas, Colorado, Géorgie,...) et une classe comprenant les Etats ayant une criminalité élevée (Alabama, Alaska, Arizona,...).

Question 2: faire de même avec $k = 2$ et commenter les profils de chaque classe.

```r
k <- 2
res2 <- kmeans(USArrests,k)
rbind(res2$centers,colMeans(USArrests))
```

```
##      Murder  Assault UrbanPop     Rape
## 1 11.857143 255.0000 67.61905 28.11429
## 2  4.841379 109.7586 64.03448 16.24828
##    7.788000 170.7600 65.54000 21.23200
```

```r
res2$cluster
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              1              1
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              1              2              1              1              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              2              2              1              2              2
```

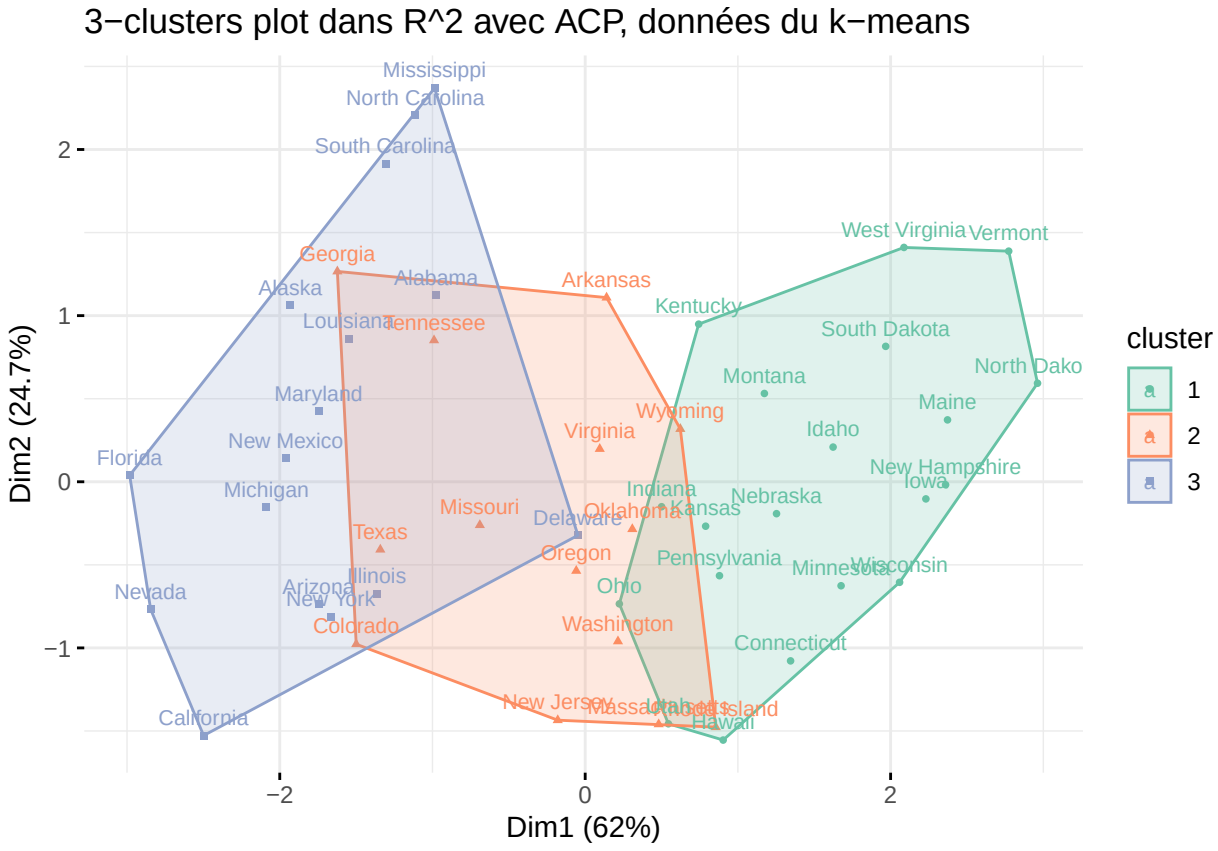
```
##          Kansas        Kentucky       Louisiana           Maine        Maryland
##               2               2               1               2               1
##   Massachusetts        Michigan       Minnesota     Mississippi        Missouri
##               2               1               2               1               2
##         Montana        Nebraska          Nevada   New Hampshire      New Jersey
##               2               2               1               2               2
##      New Mexico        New York  North Carolina    North Dakota            Ohio
##               1               1               1               2               2
##        Oklahoma          Oregon    Pennsylvania    Rhode Island  South Carolina
##               2               2               2               2               1
##    South Dakota       Tennessee           Texas            Utah         Vermont
##               2               1               1               2               2
##        Virginia      Washington   West Virginia       Wisconsin         Wyoming
##               2               2               2               2               2
```

A la vue du résultat de la fonction rbind() on constate donc que les Etats du cluster 1 sont ceux à “faible criminalité” et ceux du cluster 2 à “forte criminalité”.

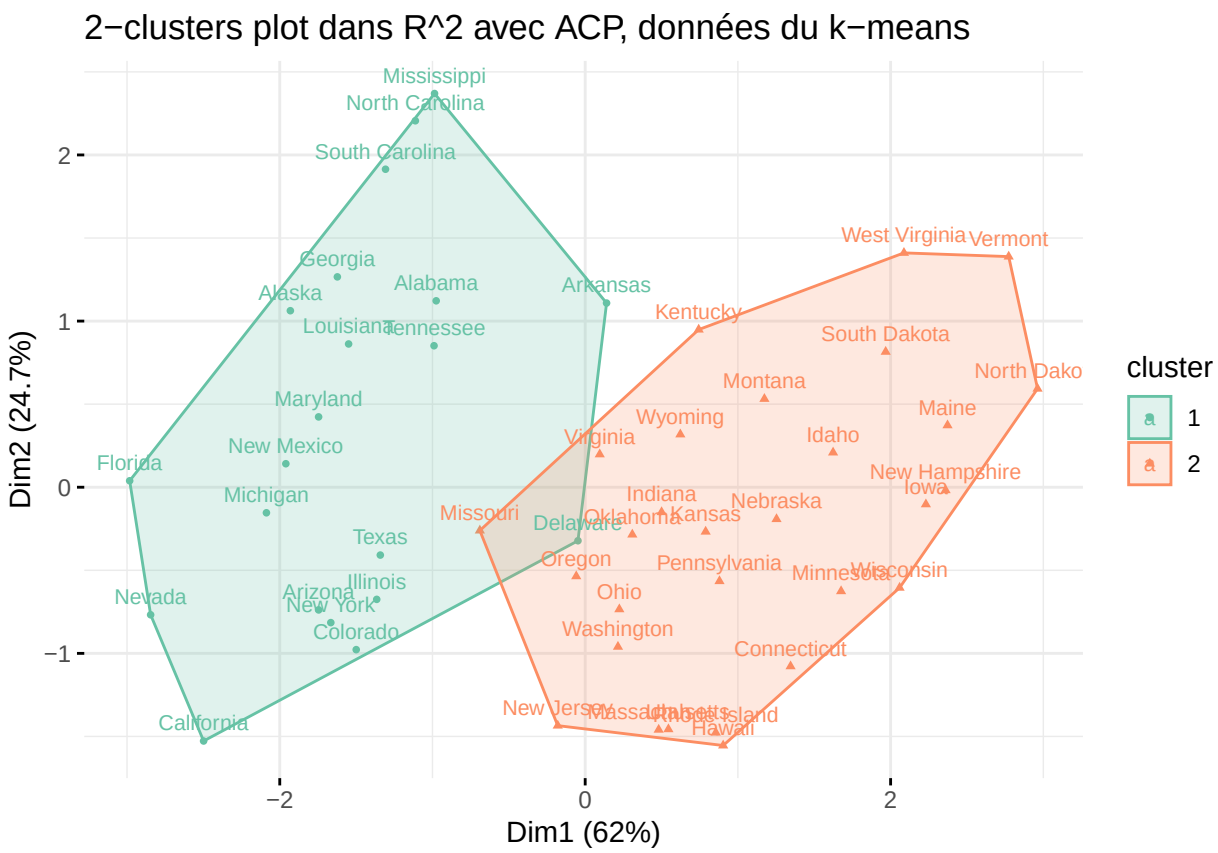
La fonction fviz_cluster() du package factoextra permet une représentation des cluster dans le plan en utilisant une ACP. (voir : https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz_cluster)

On peut ainsi représenter le clustering par k-means avec 3 et 2 clusters de la manière suivante :

```
fviz_cluster(res3, data = USArrests, palette = "Set2", ggtheme = theme_minimal(), show.clust.cent = FALS
```



```
fviz_cluster(res2, data = USArrests, palette = "Set2", ggtheme = theme_minimal(), show.clust.cent = FALS
```



Plusieurs choses sont ici intéressantes : Le passage de 3 à 2 cluster ne change rien pour les Etats situés dans les cluster 2 et 3 (rouge et bleus) dans la situation à 3 clusters. C'est le cluster 1 (vert) qui se retrouve réparti entre les 2 et 3 lors du passage à deux cluster.

Ensuite, si l'on veut rentrer plus dans le détail, dans les deux cas, le Delaware provoque un "recouvrement" plus ou moins important des cluster. Il pourrait donc être intéressant de regarder pourquoi. Ce recouvrement "visuel" peut-être est aussi dû à la représentation dans $\mathbb{R}^2$ avec une ACP à deux dimensions (qui ne compte "que" 86% de l'information), et qu'une ACP à 3 dimensions pourrait nous montrer autre chose.

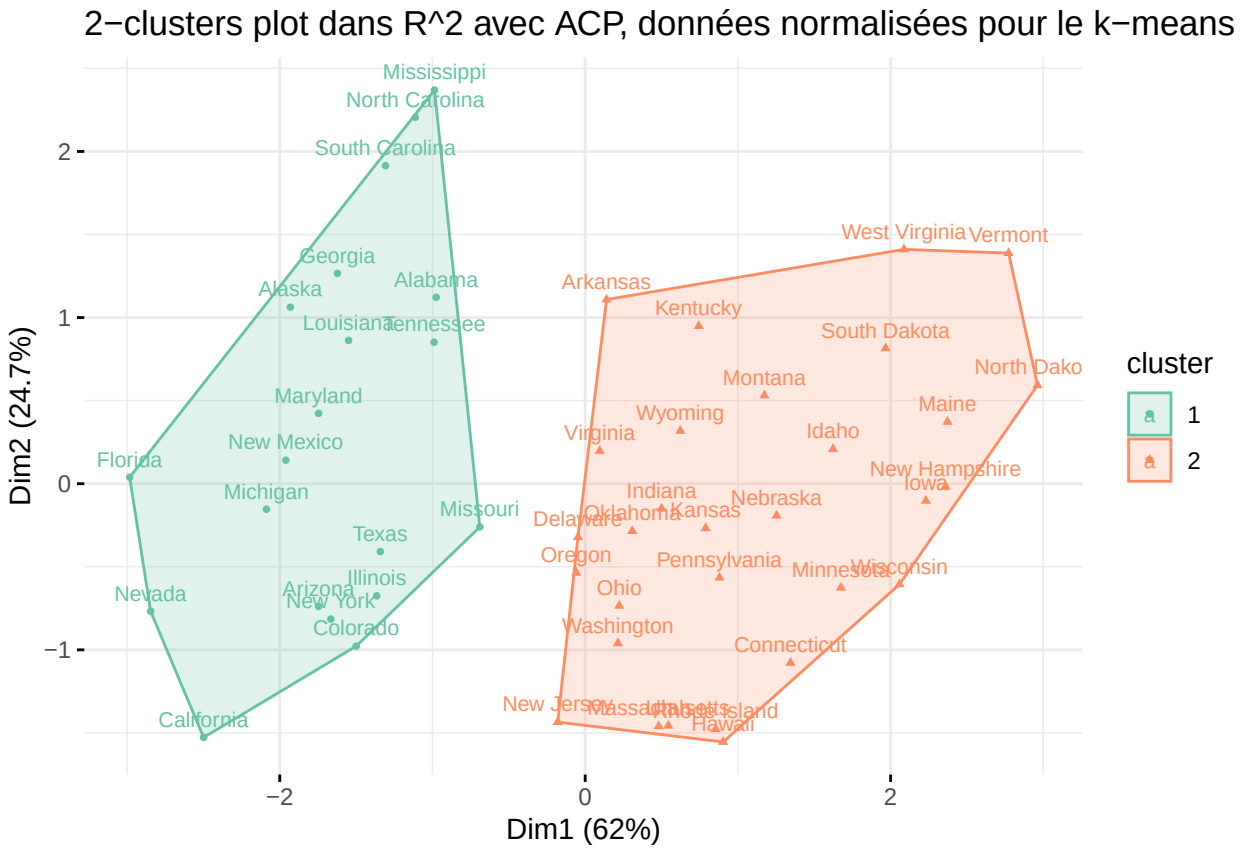
On constate en regardant en détail les valeurs du Delaware et de son plus poche voisin sur l'ACP dans $\mathbb{R}^2$, l'Oregon, respectivements classés dans les états à "forte" et "faible" criminalité par l'algorithme k-means et en comparant aux moyennes des cluster trouvés que le k-means à eu tendance à privillégier les données d'"Assault" et de "Population", alors que les donnée "Murder" et "Rape" du Delaware sont beaucoup plus proche de la moyenne des états classés à "faible criminalité" :

```
USArrests[c("Oregon","Delaware"),]
```

```
##          Murder Assault UrbanPop Rape
## Oregon      4.9     159       67 29.3
## Delaware    5.9     238       72 15.8
```

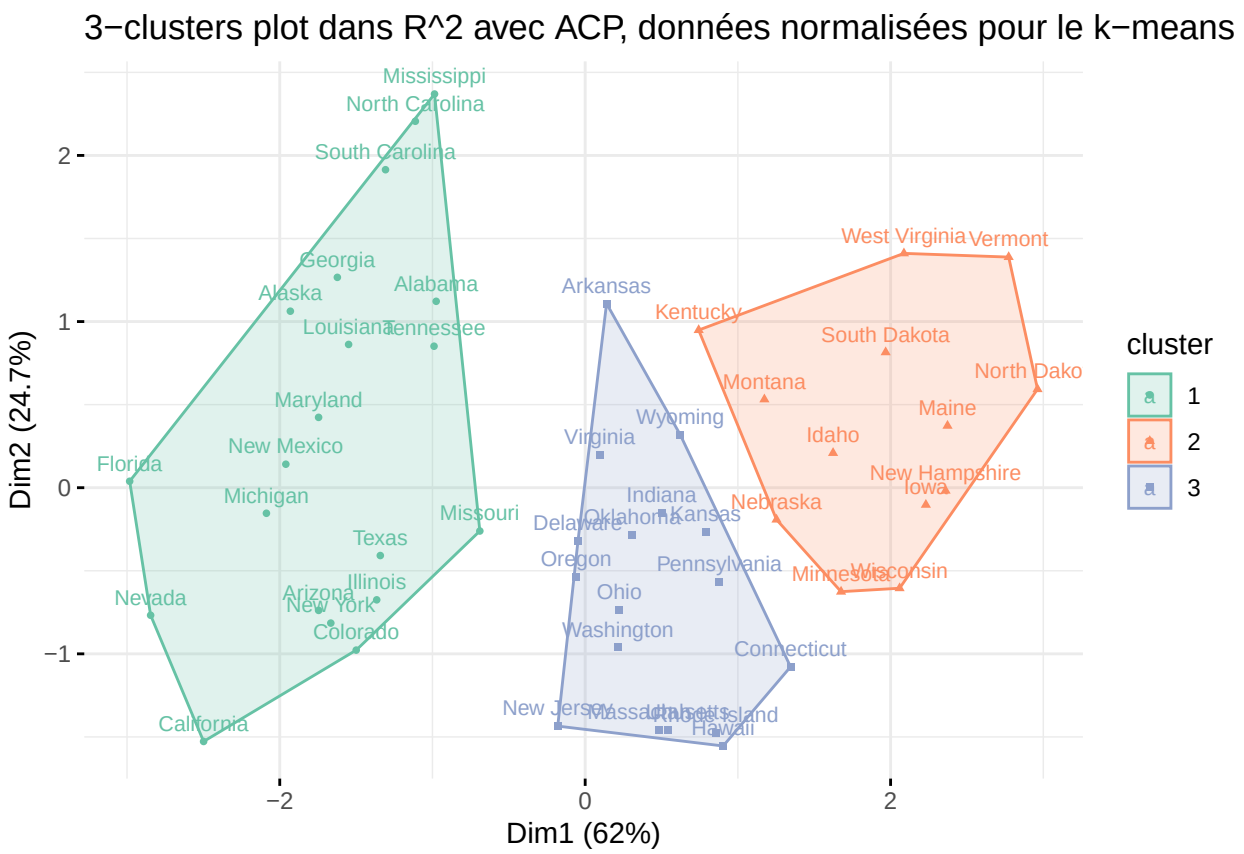
Normaliser les données avec la fonction scale() de R permet alors de "régler" ce problème (l'Arkansas a aussi basculé dans les états à "faible" criminalité). l'ACP étant normalisée et les données ne l'étant pas ci-dessus, cela explique aussi la situation du Delaware constatée plus haut :

```
res2_scale <- kmeans(scale(USArrests),k)
fviz_cluster(res2_scale, data = scale(USArrests), palette = "Set2", ggtheme = theme_minimal(), show.clus
```



On peut alors se poser la question de la normalisation pour 3 clusters :

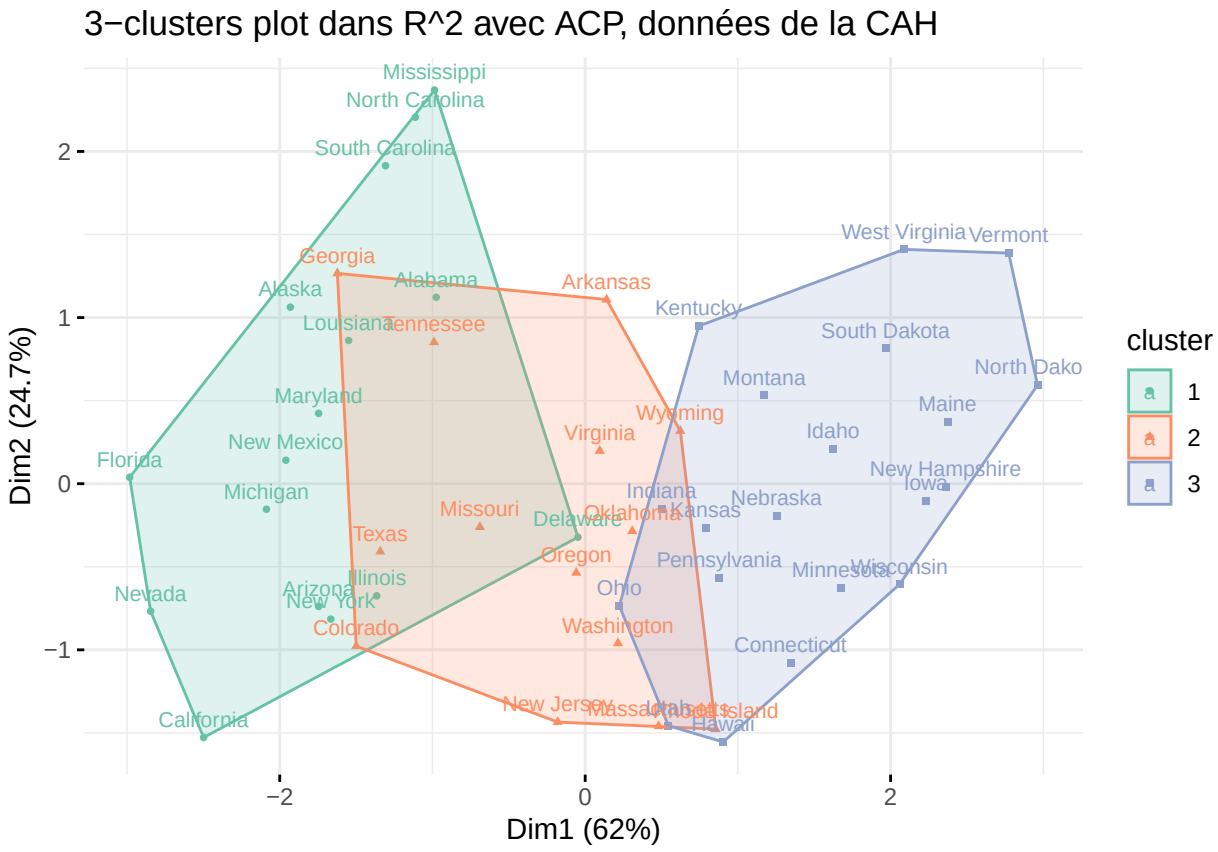
```
res3_scale <- kmeans(scale(USArrests),3)
fviz_cluster(res3_scale, data = scale(USArrests), palette = "Set2", ggtheme = theme_minimal(), show.clus
```



On constate ici une changement plus radical, mais avec des cluster plus représentatifs lorsqu'on regarde les données.

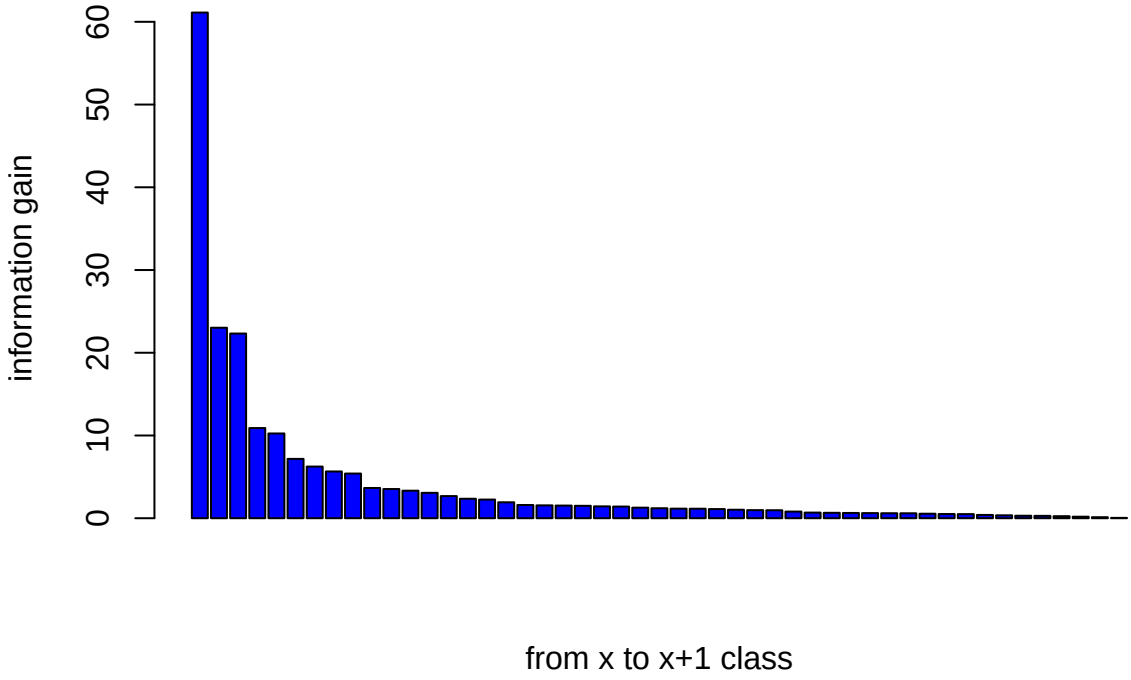
On peut aussi comparer l'algorithme k-means avec la classification CAH faite plus tôt :

```
fviz_cluster(list(data=USArrests,cluster = cutree(res.hclust,k=3)), palette = "Set2", ggtheme = theme_mi
```

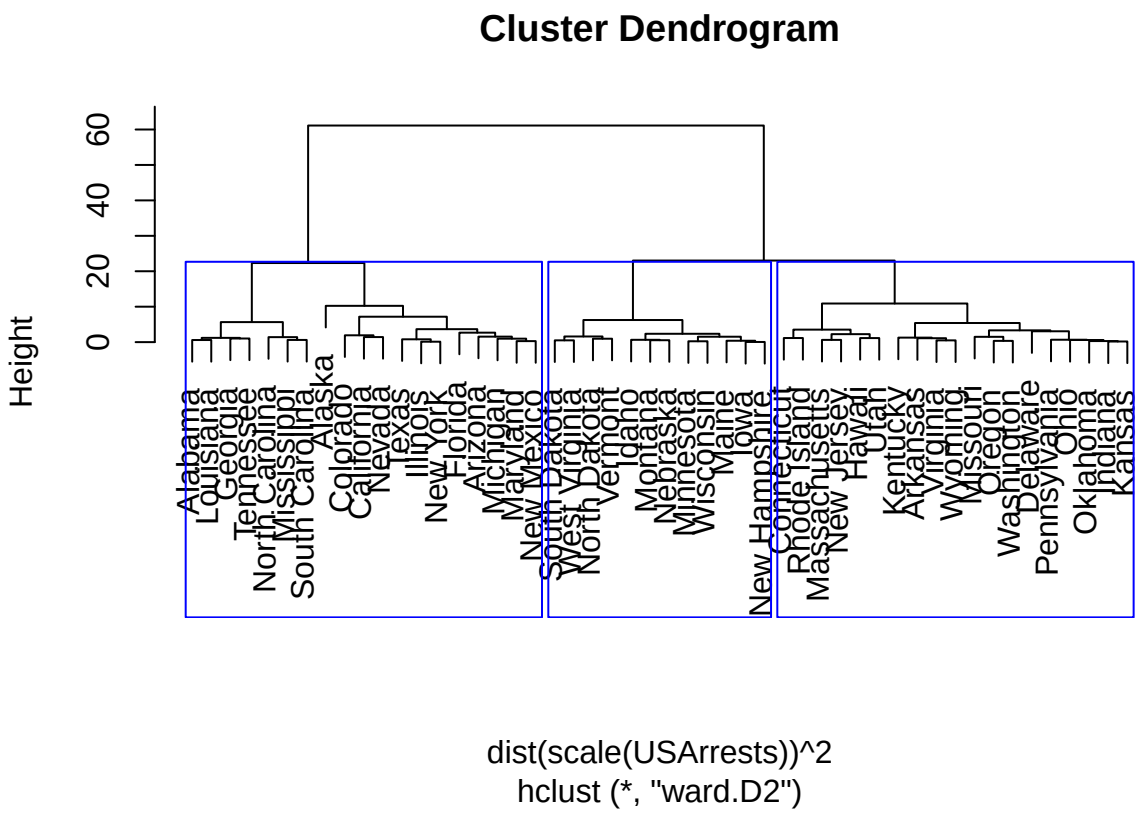


On fera la même remarque de “recouvrements” et on proposera d’effectuer la clasification CAH sur un jeu de données normalisées :

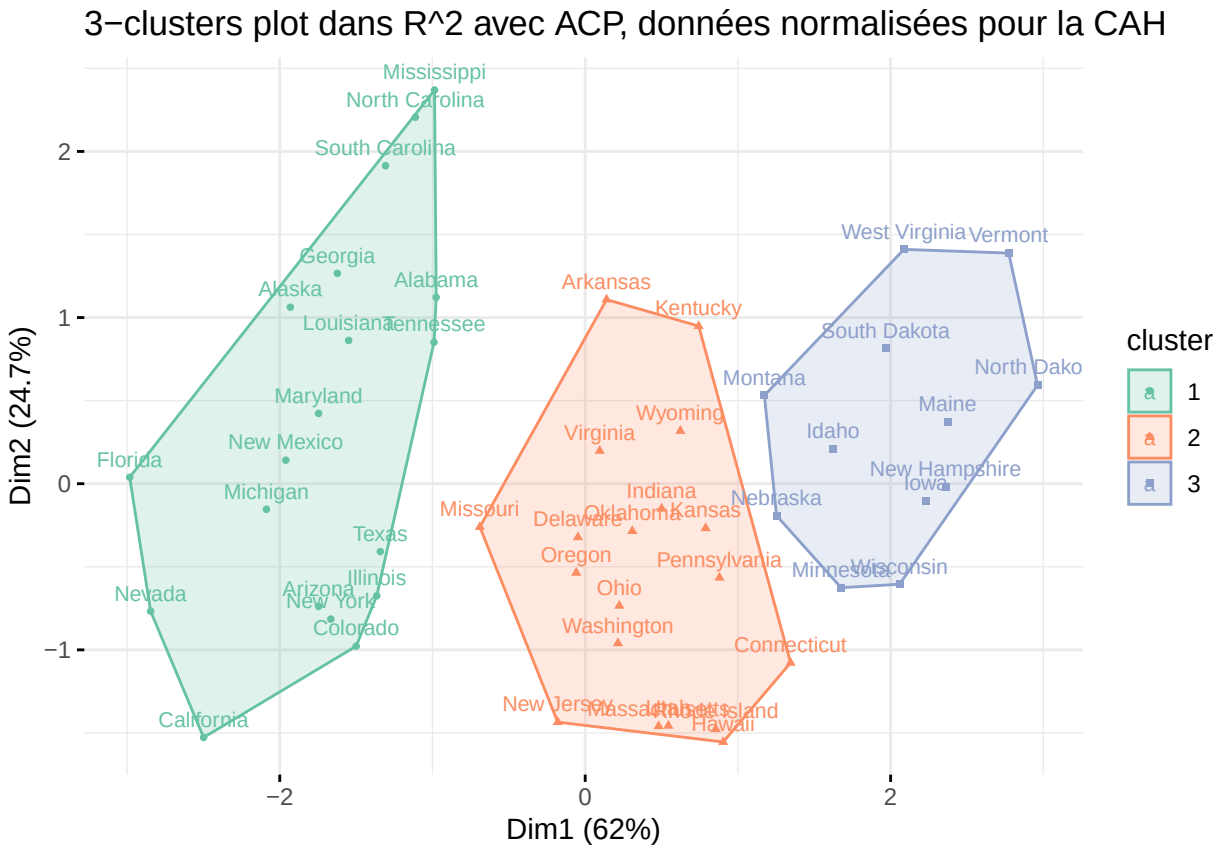
```
res.hclust.scale = hclust(dist(scale(USArrests))^2,method="ward.D2")
barplot(sort(res.hclust.scale$height, decreasing = TRUE), col = 'blue', ylab = "information gain", xlab
```



```
plot(res.hclust.scale)
rect.hclust(res.hclust.scale, 3, border = "blue")
```




```
fviz_cluster(list(data=scale(USArrests),cluster = cutree(res.hclust.scale,k=3)), palette = "Set2", ggth
```



On se rend compte que les cluster ne présenent plus de recouvrements, et seul le Kentucky est passé du cluster 1 au 2 par rapport au k-means. Ils sont sinon identiques.