TP Classification non supervisée - Correction

Analyse des données

Master ISEFAR - M1

```r
rm(list=ls())
library("tidyverse") #pour avoir de 'beaux' graphiques
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library("FactoMineR") #pour effectuer l'ACP
library("factoextra") #pour extraire et visualiser les résultats issus de FactoMineR
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

1 Criminalités aux USA

1.1 Données

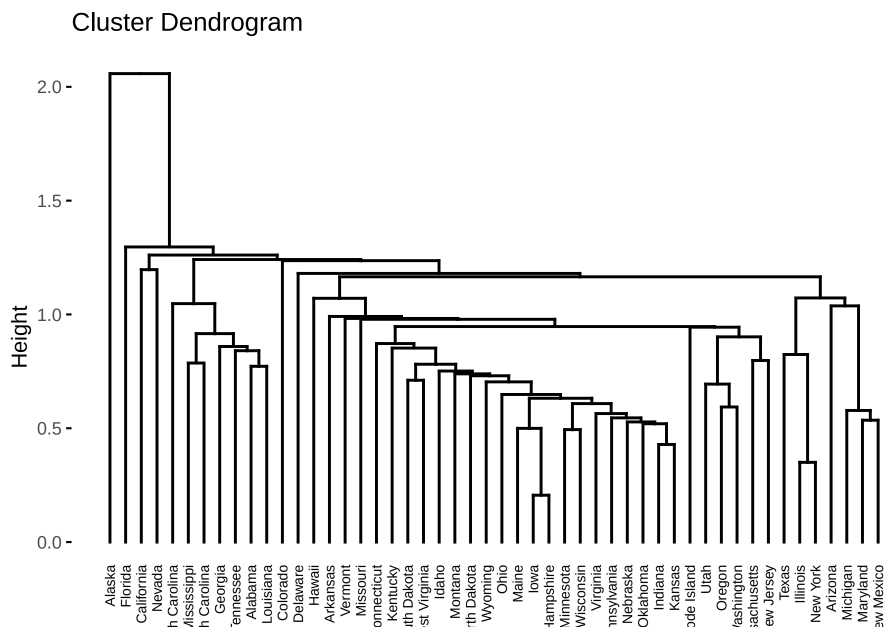
```r
data(USArrests)
```

1.2 Normalisation des données et calcul des distances entre individus avec la distance euclidienne

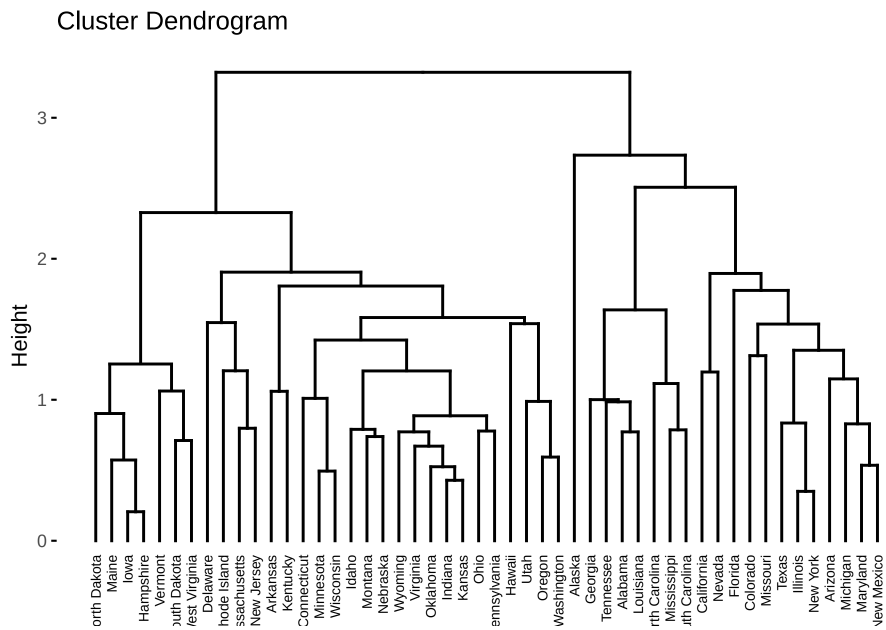
```
USArrests.cr <-  USArrests %>% scale(.,scale=TRUE, center=TRUE)
USArrests.dist <- USArrests.cr %>% dist(., method = "euclidean")
```

1.3 CAH

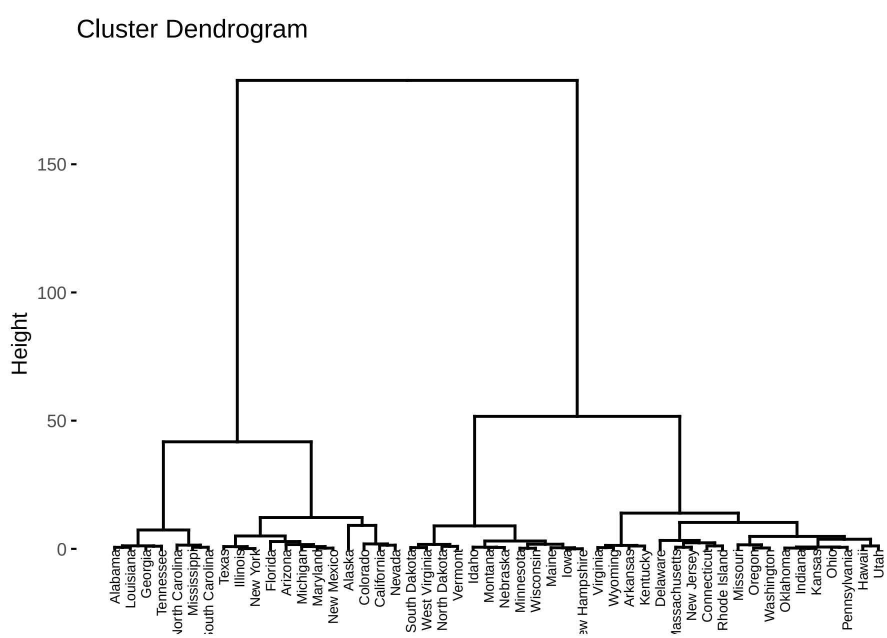
```
# Lien simple
USArrests.single<-USArrests.dist %>% hclust(., method = "single")
fviz_dend(USArrests.single, cex = 0.5)
```



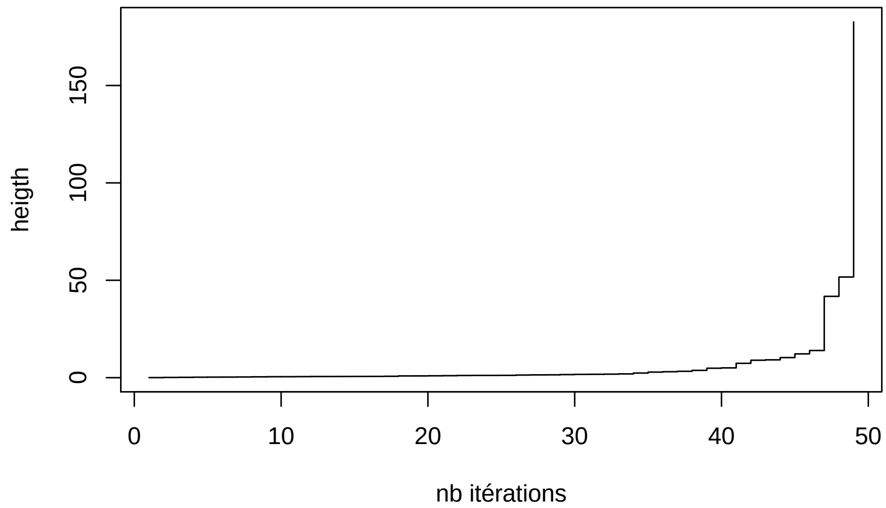

```
# Lien complet
USArrests.average<-USArrests.dist %>% hclust(., method = "average")
fviz_dend(USArrests.average, cex = 0.5)
```




```r
# distance de ward
USArrests.ward<-USArrests.dist^2 %>% hclust(., method = "ward.D")
fviz_dend(USArrests.ward, cex = 0.5)
```

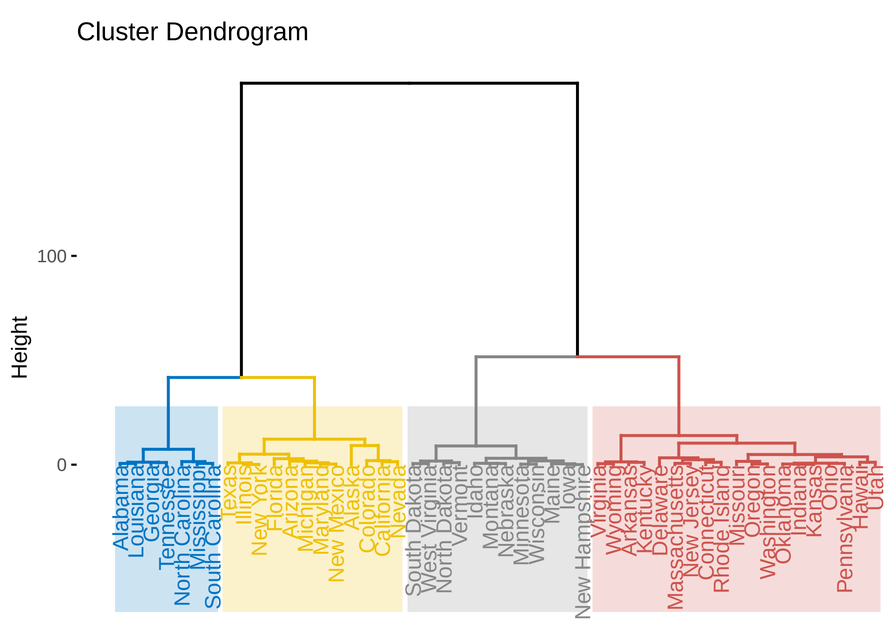



```r
# Courbe de l'augmentation de l'inertie intra-groupe en fonction du nombre d'itérations
plot(USArrests.ward$height,type="s",xlab="nb itérations",ylab="heigth")
```



On choisit 4 groupes

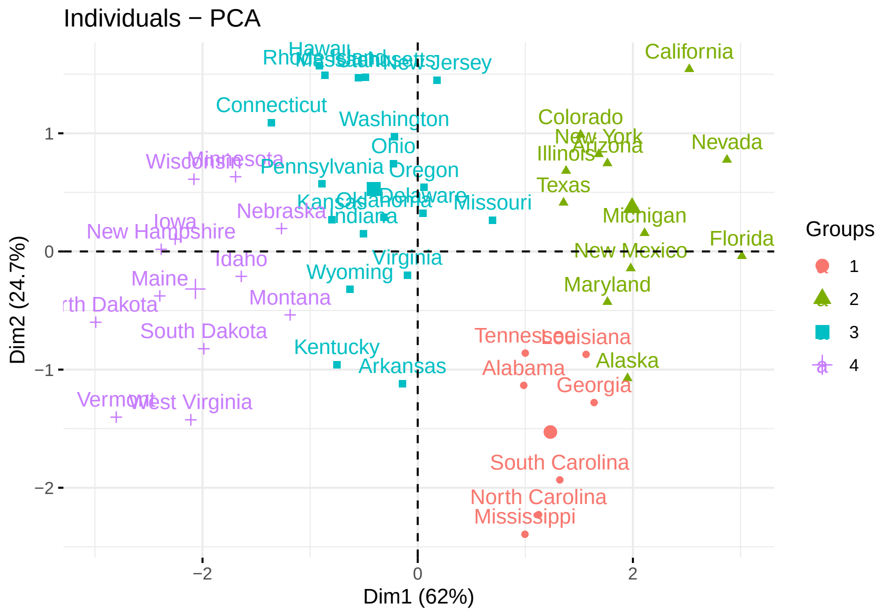
```r
fviz_dend(USArrests.ward,
          k=4,
          cex = 0.8,
          palette="jco",
          rect = TRUE, rect_fill = TRUE, # Rectangle autour des groupes
          rect_border = "jco",
          labels_track_height = 70
          )
```



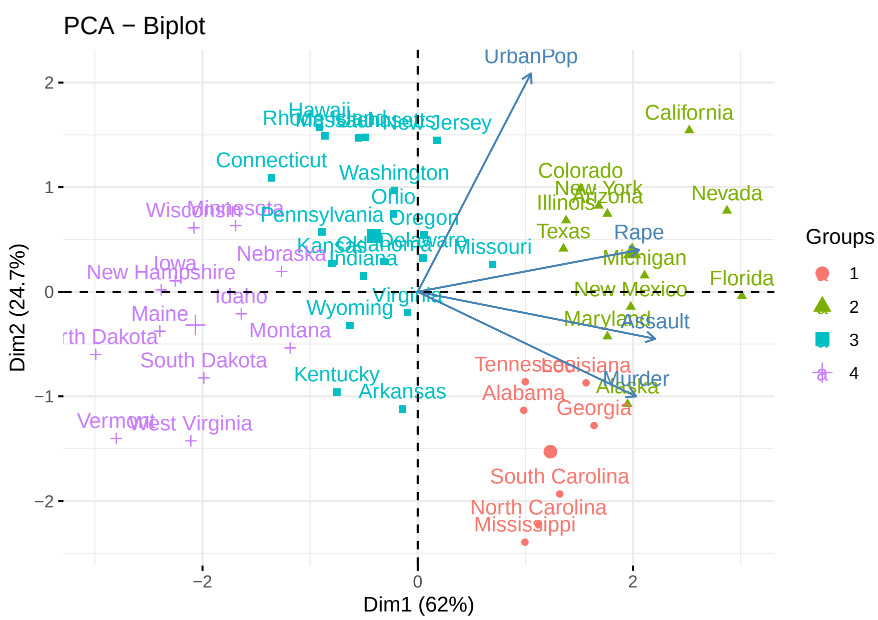
1.4 Représentation des groupes sur le plan principal de l'ACP

```r
# On récupère les k groupes
cluster.CAH <- USArrests.ward %>%  cutree(., k =4)
```

```
# ACP
res.pca=PCA(USArrests,scale.unit = TRUE,ncp = 4,graph=FALSE)
# visualiser les classes sur le premier plan factoriel de l'ACP
fviz_pca_ind(res.pca,axes=c(1,2),habillage=as.factor(cluster.CAH))
```




```
fviz_pca_biplot(res.pca,axes=c(1,2),habillage=as.factor(cluster.CAH))
```

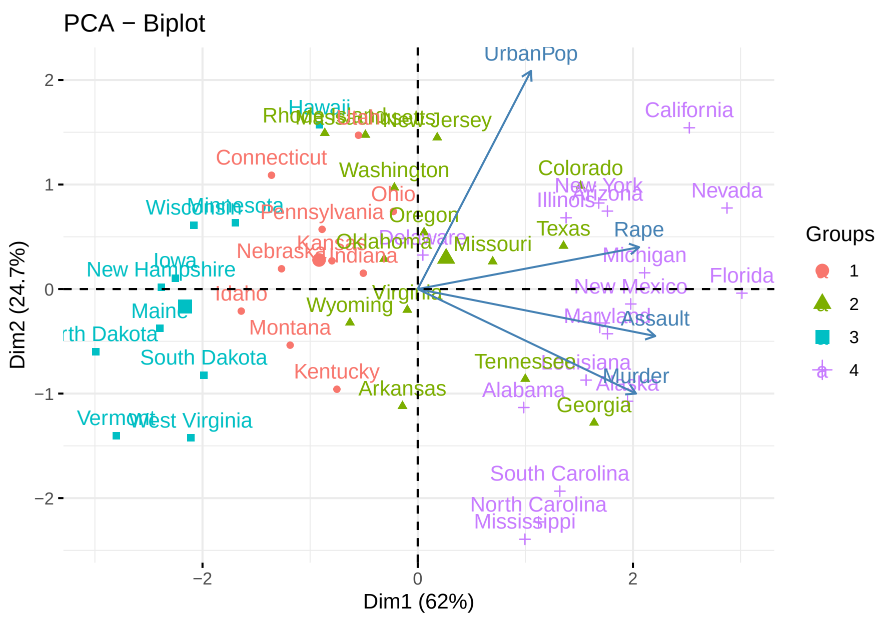



```
# moyennes des variables par groupe
knitr::kable(aggregate(USArrests, by=list(as.factor(cluster.CAH)),mean),digits=1)
```

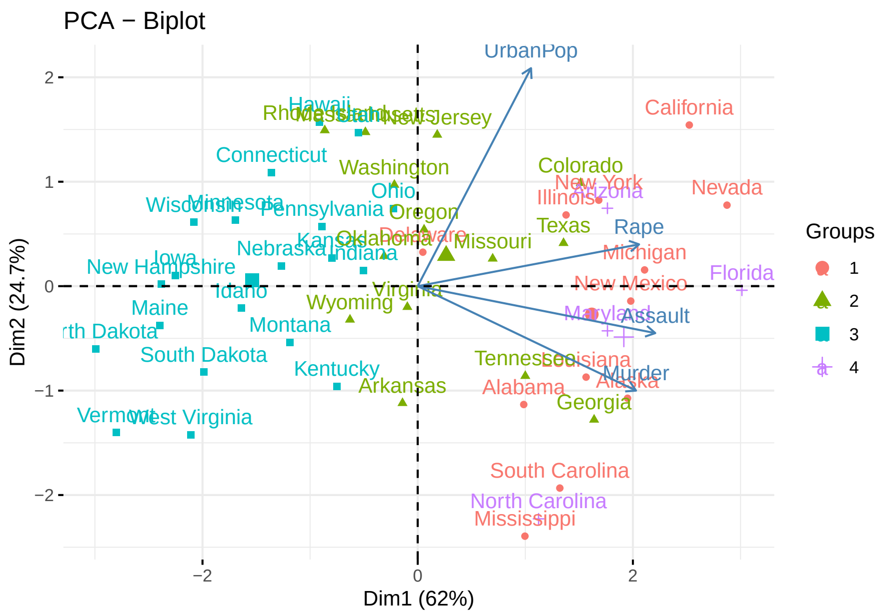
Group.1	Murder	Assault	UrbanPop	Rape
1	14.7	251.3	54.3	21.7
2	11.0	264.0	76.5	33.6
3	6.2	142.1	71.3	19.2
4	3.1	76.0	52.1	11.8

1.5 K-means

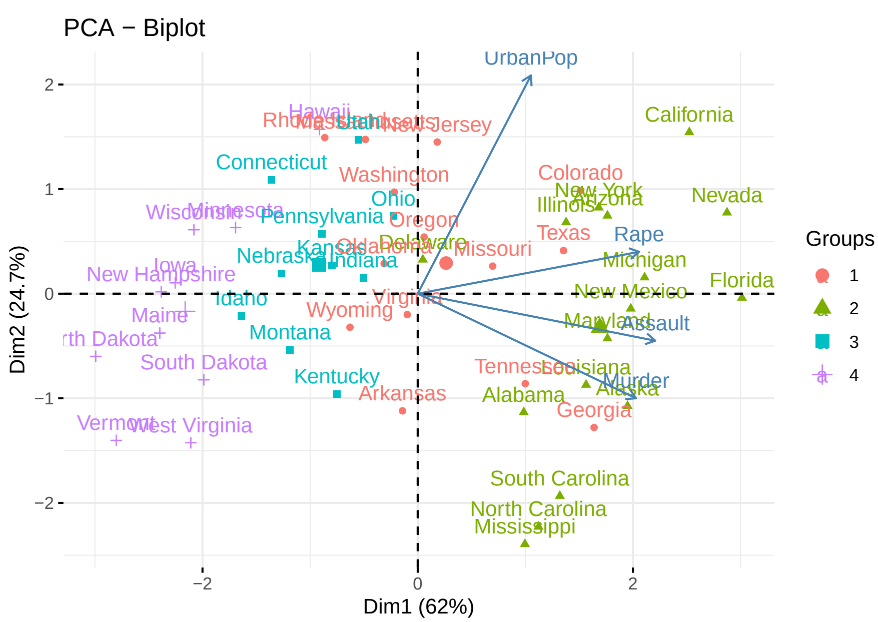
```
# 1 fois
res.kmeans <- USArrests %>% kmeans(.,centers =4,nstart = 1)
cluster.kmeans <- res.kmeans$cluster
fviz_pca_biplot(res.pca,axes=c(1,2),habillage=as.factor(cluster.kmeans))
```




```
# 1 fois
res.kmeans <- USArrests %>% kmeans(.,centers =4,nstart = 1)
cluster.kmeans <- res.kmeans$cluster
fviz_pca_biplot(res.pca,axes=c(1,2),habillage=as.factor(cluster.kmeans))
```




```r
# 10 fois
res.kmeans <- USArrests %>% kmeans(.,centers =4,nstart = 10)
cluster.kmeans <- res.kmeans$cluster
fviz_pca_biplot(res.pca,axes=c(1,2),habillage=as.factor(cluster.kmeans))
```



1.6 Comparaisons CAH et k-means

```r
knitr::kable(cbind(cluster.kmeans,cluster.CAH))
```

	cluster.kmeans	cluster.CAH
Alabama	2	1
Alaska	2	2
Arizona	2	2
Arkansas	1	3
California	2	2
Colorado	1	2
Connecticut	3	3
Delaware	2	3
Florida	2	2
Georgia	1	1
Hawaii	4	3
Idaho	3	4
Illinois	2	2
Indiana	3	3
Iowa	4	4
Kansas	3	3
Kentucky	3	3
Louisiana	2	1
Maine	4	4
Maryland	2	2
Massachusetts	1	3
Michigan	2	2
Minnesota	4	4
Mississippi	2	1
Missouri	1	3
Montana	3	4
Nebraska	3	4
Nevada	2	2
New Hampshire	4	4
New Jersey	1	3
New Mexico	2	2
New York	2	2
North Carolina	2	1
North Dakota	4	4
Ohio	3	3
Oklahoma	1	3
Oregon	1	3
Pennsylvania	3	3
Rhode Island	1	3
South Carolina	2	1
South Dakota	4	4
Tennessee	1	1
Texas	1	2
Utah	3	3
Vermont	4	4
Virginia	1	3
Washington	1	3
West Virginia	4	4
Wisconsin	4	4
Wyoming	1	3

2 Fertilité et indicateurs socio-économiques en Suisse

2.1 Données

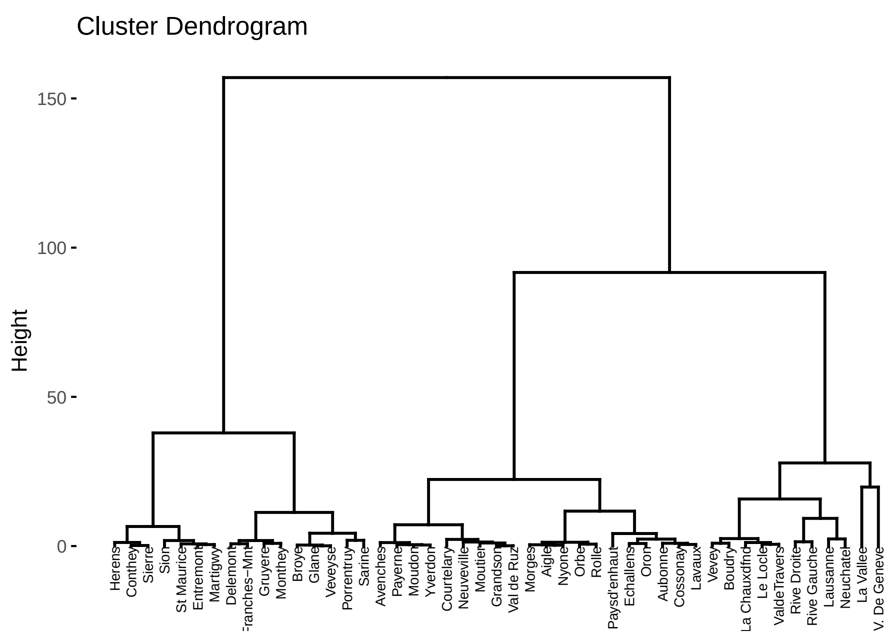
```r
data(swiss)
```

2.2 CAH avec la distance de Ward

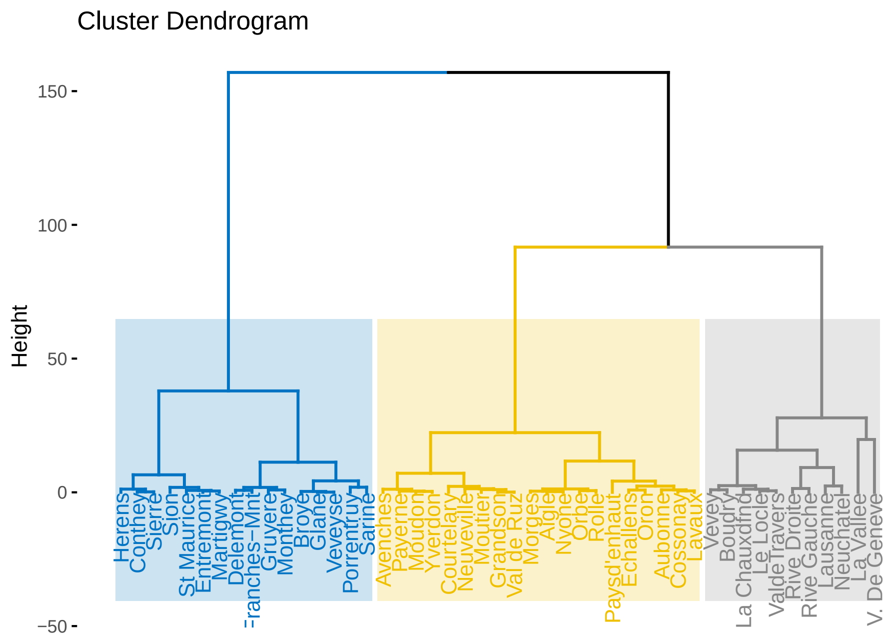
```r
# CAH
swiss.var <- swiss %>% dplyr::select(-Fertility)

swiss.cr <- swiss.var %>% scale(.,scale=TRUE, center=TRUE)
swiss.dist <- swiss.cr %>% dist(., method = "euclidean")
swiss.ward<- swiss.dist^2 %>%hclust(., method = "ward.D")

# Nombre de groupes?
fviz_dend(swiss.ward, cex = 0.5)
```

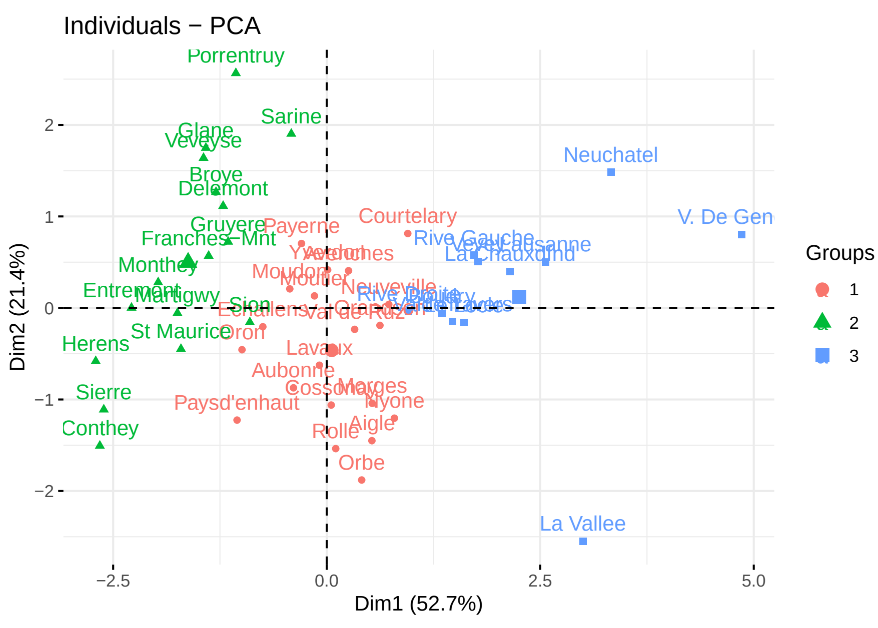



```r
fviz_dend(swiss.ward,
          k=3,
          cex = 0.8,
          palette="jco",
          rect = TRUE, rect_fill = TRUE, # Rectangle autour des groupes
          rect_border = "jco",
          labels_track_height = 40
          )
```

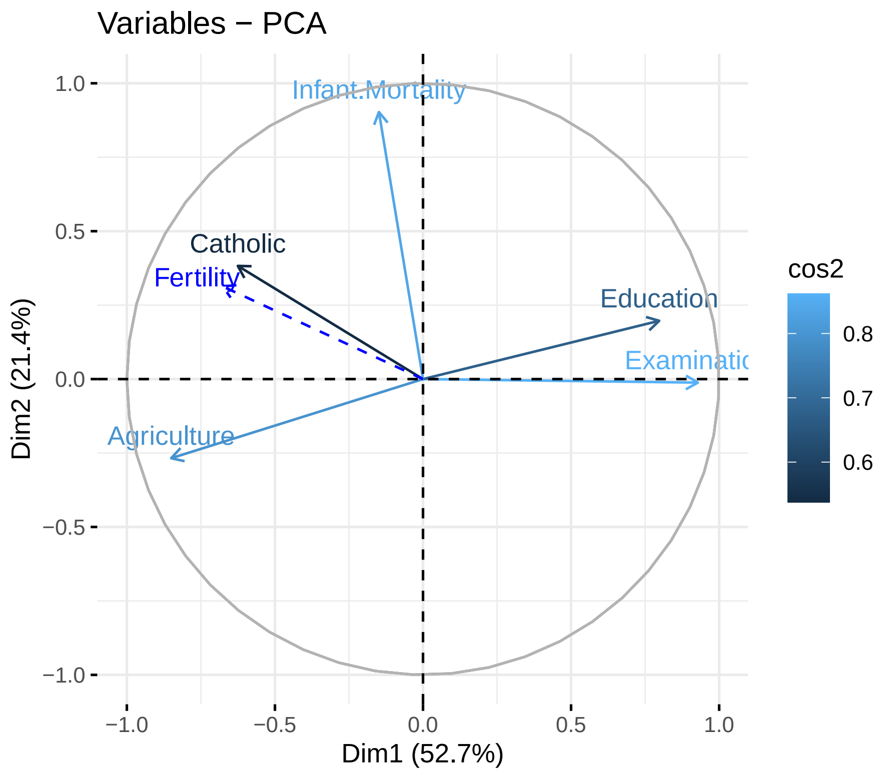


2.3 Représentation des groupes à l'aide de l'ACP

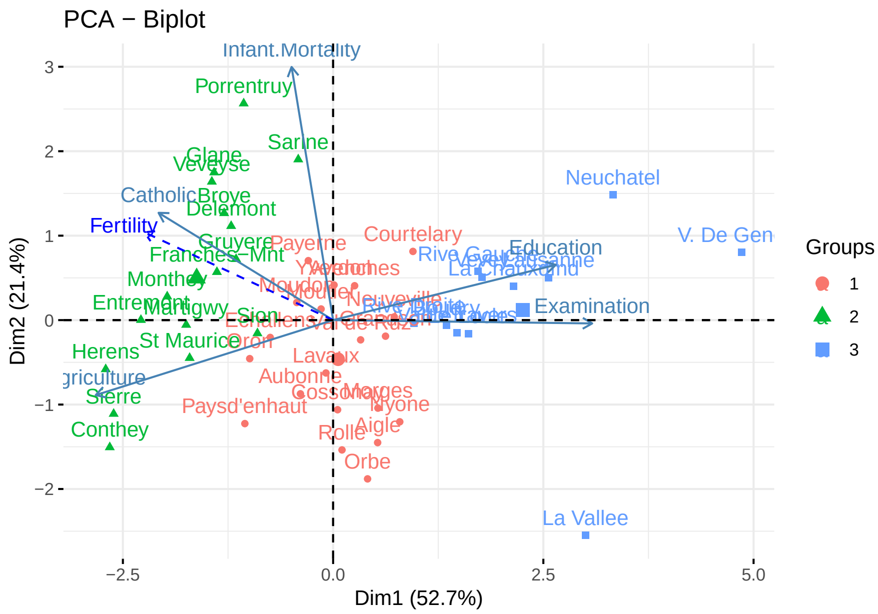
```r
cluster <- swiss.ward %>% cutree(., k =3)
# ACP
res.pca=PCA(swiss,scale.unit = TRUE,ncp = 5,graph=FALSE,quanti.sup = 1)
# visualiser les classes sur le premier plan factoriel de l'ACP
fviz_pca_ind(res.pca,axes=c(1,2),habillage=as.factor(cluster))
```




```r
fviz_pca_var(res.pca,axes=c(1,2),col.var="cos2")
```




```r
fviz_pca_biplot(res.pca,axes=c(1,2),habillage=as.factor(cluster))
```




```r
# moyennes des variables par groupe
knitr::kable(aggregate(swiss[,-1], by=list(as.factor(cluster)),mean),digits=1)
```

Group.1	Agriculture	Examination	Education	Catholic	Infant.Mortality
1	54.8	16.5	7.8	7.8	19.8
2	65.5	9.4	6.6	96.2	20.8
3	21.5	26.7	23.0	21.8	19.1

3 Décathlon

3.1 Données et statistiques simples

```
# Données
data(decathlon)
decathlon <- decathlon %>% filter(Competition=="OlympicG") %>% dplyr::select(-Competition)
knitr::kable(head(decathlon))
```

	100m	Long.jump	Shot.put	High.jump	400m	110m.hurdle	Discus	Pole.vault	Javeline	1500m	Rank	Points
Sebrle	10.85	7.84	16.36	2.12	48.36	14.05	48.72	5.0	70.52	280.01	1	8893
Clay	10.44	7.96	15.23	2.06	49.19	14.13	50.11	4.9	69.71	282.00	2	8820
Karpov	10.50	7.81	15.93	2.09	46.81	13.97	51.65	4.6	55.54	278.11	3	8725
Macey	10.89	7.47	15.73	2.15	48.97	14.56	48.34	4.4	58.46	265.42	4	8414
Warners	10.62	7.74	14.48	1.97	47.97	14.01	43.73	4.9	55.39	278.05	5	8343
Zsivoczky	10.91	7.14	15.31	2.12	49.40	14.95	45.62	4.7	63.45	269.54	6	8287

```
dim(decathlon)
```

```
## [1] 28 12
```

```
#Statistiques simples
summary(decathlon)
```

```
##       100m          Long.jump        Shot.put       High.jump    
##  Min.   :10.44   Min.   :6.610   Min.   :13.07   Min.   :1.850  
##  1st Qu.:10.84   1st Qu.:7.020   1st Qu.:13.98   1st Qu.:1.933  
##  Median :10.90   Median :7.280   Median :14.79   Median :1.940  
##  Mean   :10.92   Mean   :7.266   Mean   :14.62   Mean   :1.976  
##  3rd Qu.:11.08   3rd Qu.:7.482   3rd Qu.:15.17   3rd Qu.:2.038  
##  Max.   :11.36   Max.   :7.960   Max.   :16.36   Max.   :2.150  
##       400m        110m.hurdle        Discus        Pole.vault   
##  Min.   :46.81   Min.   :13.97   Min.   :39.83   Min.   :4.200  
##  1st Qu.:48.93   1st Qu.:14.20   1st Qu.:42.01   1st Qu.:4.500  
##  Median :49.37   Median :14.40   Median :44.51   Median :4.700  
##  Mean   :49.61   Mean   :14.55   Mean   :44.38   Mean   :4.732  
##  3rd Qu.:50.36   3rd Qu.:14.95   3rd Qu.:45.73   3rd Qu.:4.925  
##  Max.   :53.20   Max.   :15.39   Max.   :51.65   Max.   :5.400  
##     Javeline         1500m            Rank           Points    
##  Min.   :50.62   Min.   :263.1   Min.   : 1.00   Min.   :7404  
##  1st Qu.:55.36   1st Qu.:270.7   1st Qu.: 7.75   1st Qu.:7886  
##  Median :58.94   Median :276.3   Median :14.50   Median :8022  
```

```
##  Mean   :58.95   Mean   :277.6   Mean   :14.50   Mean   :8052
##  3rd Qu.:61.00   3rd Qu.:280.4   3rd Qu.:21.25   3rd Qu.:8236
##  Max.   :70.52   Max.   :317.0   Max.   :28.00   Max.   :8893
```

```
decathlon.active <- decathlon %>% dplyr::select(-Rank,-Points)
decathlon.active %>% summarise_all(mean)
```

```
##       100m Long.jump Shot.put High.jump  400m 110m.hurdle   Discus Pole.vault
## 1 10.91571  7.265714   14.625  1.976429 49.61    14.55357 44.37571   4.732143
##   Javeline     1500m
## 1 58.94893 277.5507
```

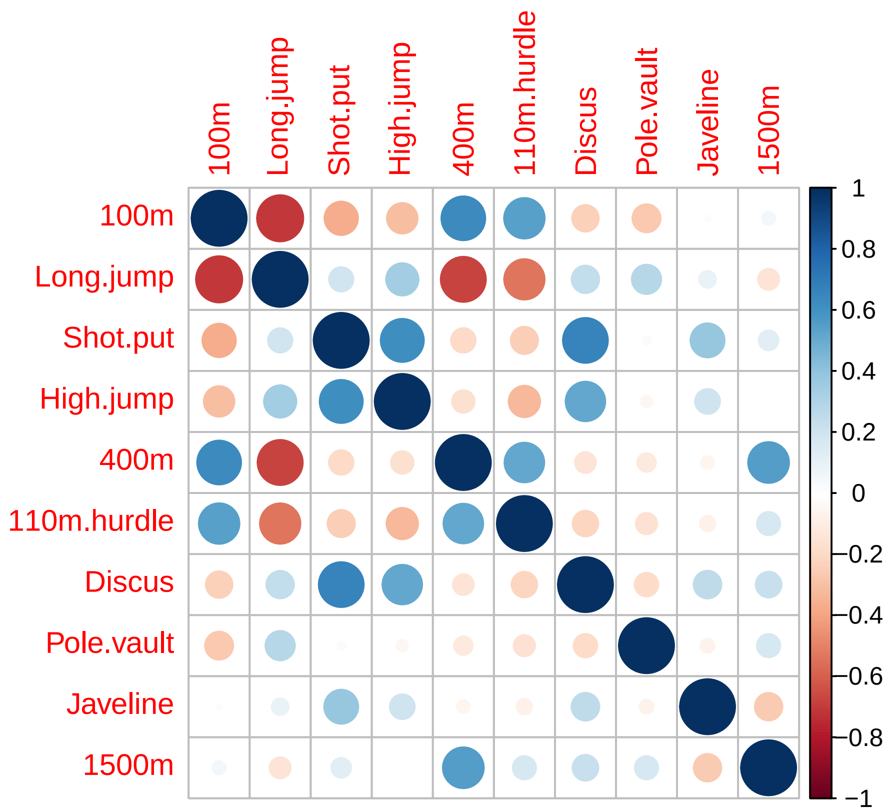
```
decathlon.active %>% summarise_all(var)
```

```
##         100m Long.jump  Shot.put   High.jump     400m 110m.hurdle   Discus
## 1 0.05337354 0.1163735 0.7331148 0.008090476 1.608978   0.1959794 10.88727
##   Pole.vault Javeline    1500m
## 1 0.08374339 24.75908 128.1838
```

```
#Corrélation
correlation <- decathlon.active %>% cor(.)
print(correlation,digits=3)
```

```
##                100m Long.jump Shot.put High.jump    400m 110m.hurdle Discus
## 100m         1.0000   -0.7050  -0.3697   -0.3093  0.6348      0.5426 -0.233
## Long.jump   -0.7050    1.0000   0.1955    0.3457 -0.6711     -0.5382  0.250
## Shot.put    -0.3697    0.1955   1.0000    0.6126 -0.1993     -0.2451  0.666
## High.jump   -0.3093    0.3457   0.6126    1.0000 -0.1692     -0.3260  0.517
## 400m         0.6348   -0.6711  -0.1993   -0.1692  1.0000      0.5199 -0.144
## 110m.hurdle  0.5426   -0.5382  -0.2451   -0.3260  0.5199      1.0000 -0.217
## Discus      -0.2333    0.2499   0.6658    0.5170 -0.1442     -0.2169  1.000
## Pole.vault  -0.2605    0.2851   0.0237   -0.0424 -0.1154     -0.1510 -0.184
## Javeline    -0.0117    0.0938   0.3833    0.2045 -0.0547     -0.0798  0.255
## 1500m        0.0584   -0.1474   0.1295   -0.0035  0.5512      0.1790  0.220
##             Pole.vault Javeline   1500m
## 100m           -0.2605  -0.0117  0.0584
## Long.jump       0.2851   0.0938 -0.1474
## Shot.put        0.0237   0.3833  0.1295
## High.jump      -0.0424   0.2045 -0.0035
## 400m           -0.1154  -0.0547  0.5512
## 110m.hurdle    -0.1510  -0.0798  0.1790
## Discus         -0.1842   0.2549  0.2202
## Pole.vault      1.0000  -0.0661  0.1795
## Javeline       -0.0661   1.0000 -0.2515
## 1500m           0.1795  -0.2515  1.0000
```

```
correlation %>% corrplot
```



3.2 ACP

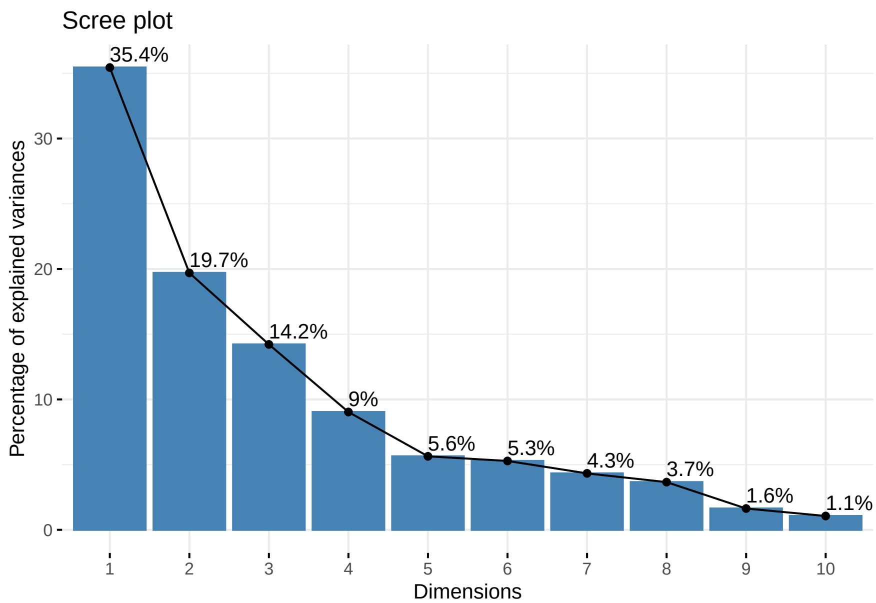
```r
res.pca=PCA(decathlon,scale.unit = TRUE,ncp = 10,quanti.sup = 11:12,graph=FALSE)
# les variables supplémentaires sont intégrées au graphe mais ne sont pas
# prises en compte pour l'ACP
```

3.2.1 Valeurs propres

```r
res.pca$eig
```

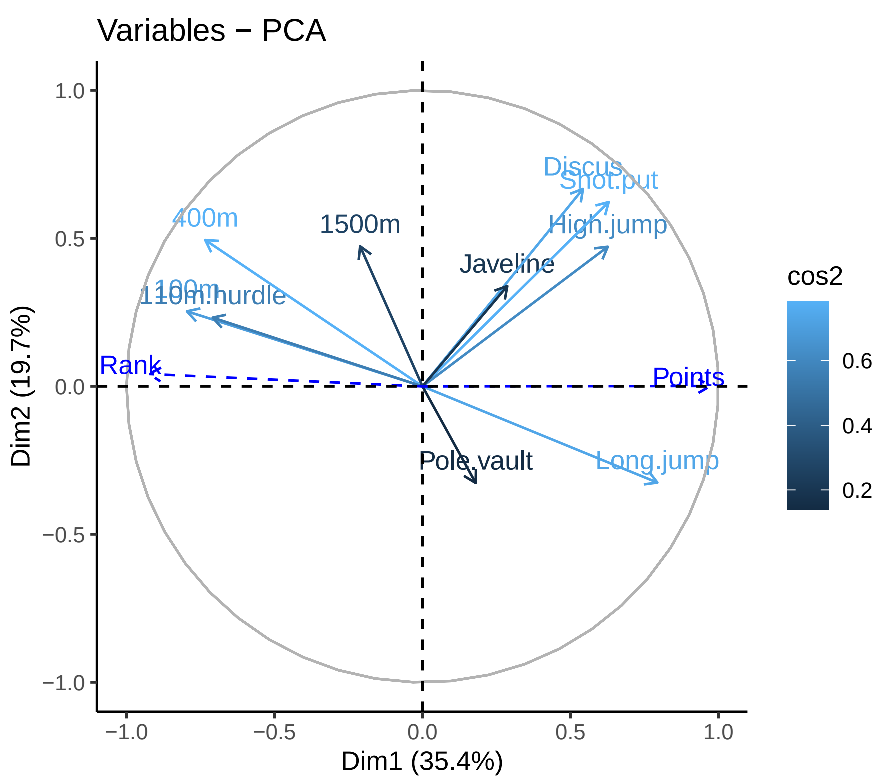
```
##         eigenvalue percentage of variance cumulative percentage of variance
## comp 1   3.5446573              35.446573                          35.44657
## comp 2   1.9699560              19.699560                          55.14613
## comp 3   1.4217248              14.217248                          69.36338
## comp 4   0.9034912               9.034912                          78.39829
## comp 5   0.5636320               5.636320                          84.03461
## comp 6   0.5282270               5.282270                          89.31688
## comp 7   0.4328613               4.328613                          93.64550
## comp 8   0.3658102               3.658102                          97.30360
## comp 9   0.1634956               1.634956                          98.93855
## comp 10  0.1061447               1.061447                         100.00000
```

```r
fviz_eig(res.pca, addlabels = TRUE)
```

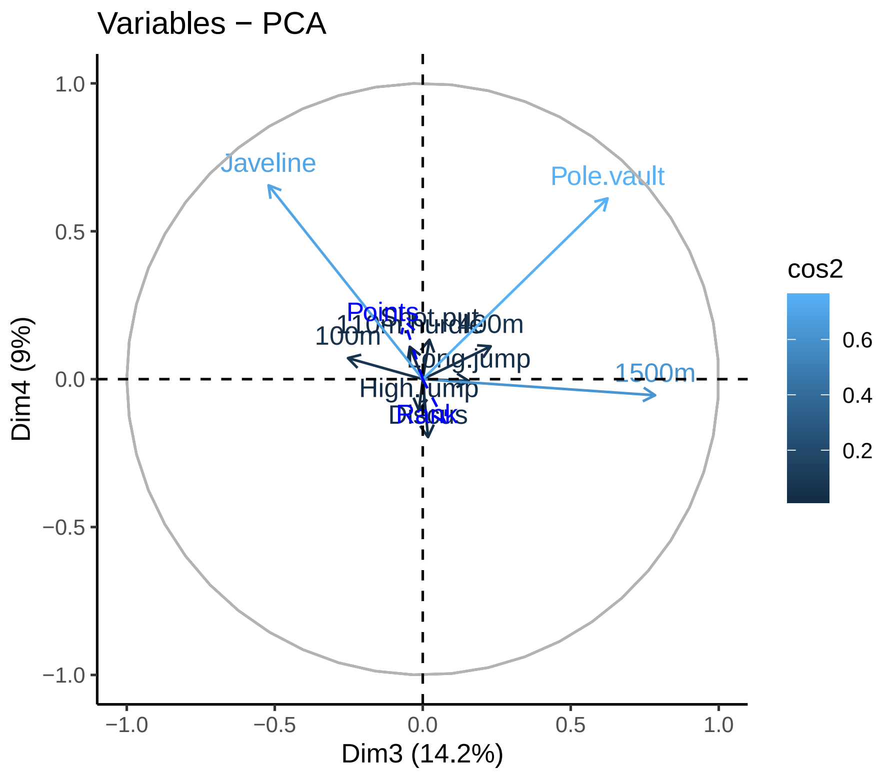


3.2.2 Variables

```r
fviz_pca_var(res.pca, geom = c("text", "arrow"), col.var = "cos2",axes=1:2) + theme_classic()
```

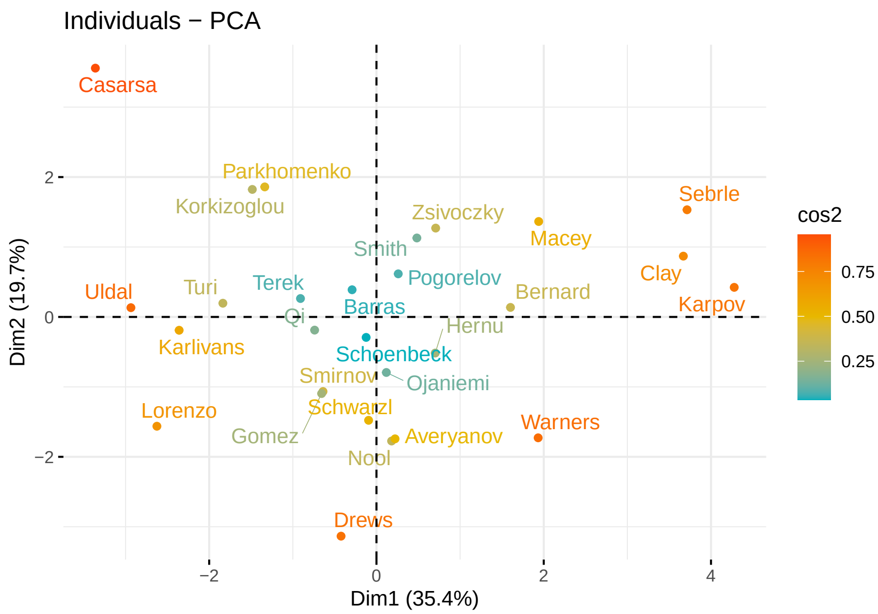


```r
fviz_pca_var(res.pca, geom = c("text", "arrow"), col.var = "cos2",axes=3:4) + theme_classic()
```

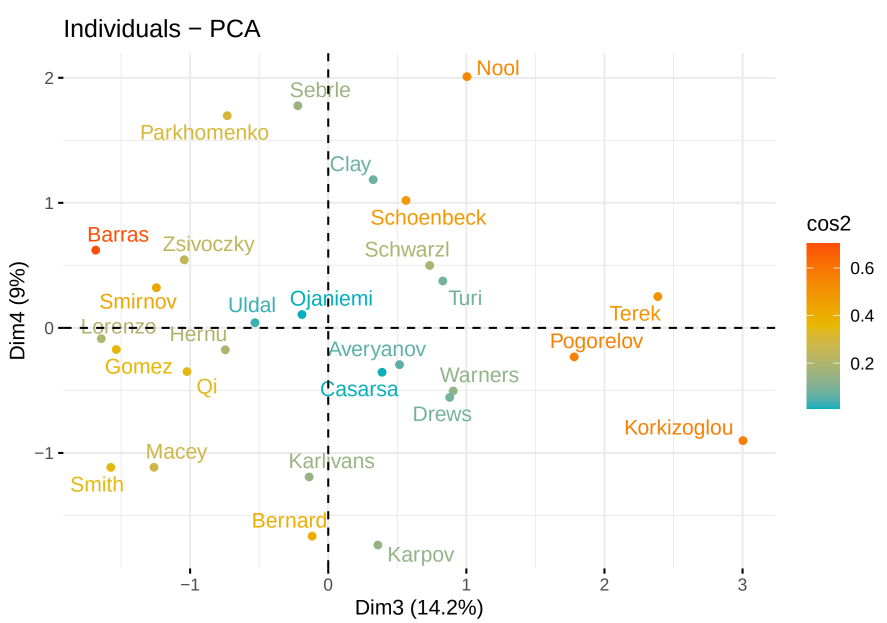


3.2.3 Individus

```r
fviz_pca_ind (res.pca, col.ind = "cos2",axes=1:2,
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE # Évite le chevauchement de texte
              )
```

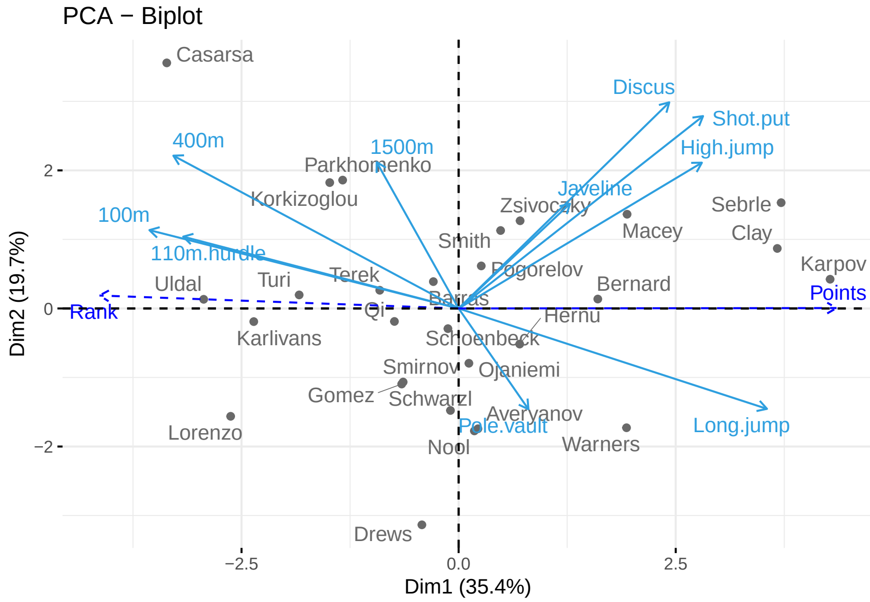



```r
fviz_pca_ind (res.pca, col.ind = "cos2",axes=3:4,
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE # Évite le chevauchement de texte
              )
```



3.2.4 Biplot

```r
fviz_pca_biplot(res.pca, repel = TRUE,
                col.var = "#2E9FDF", # Couleur des variables
                col.ind = "#696969"  # Couleur des individus
                )
```

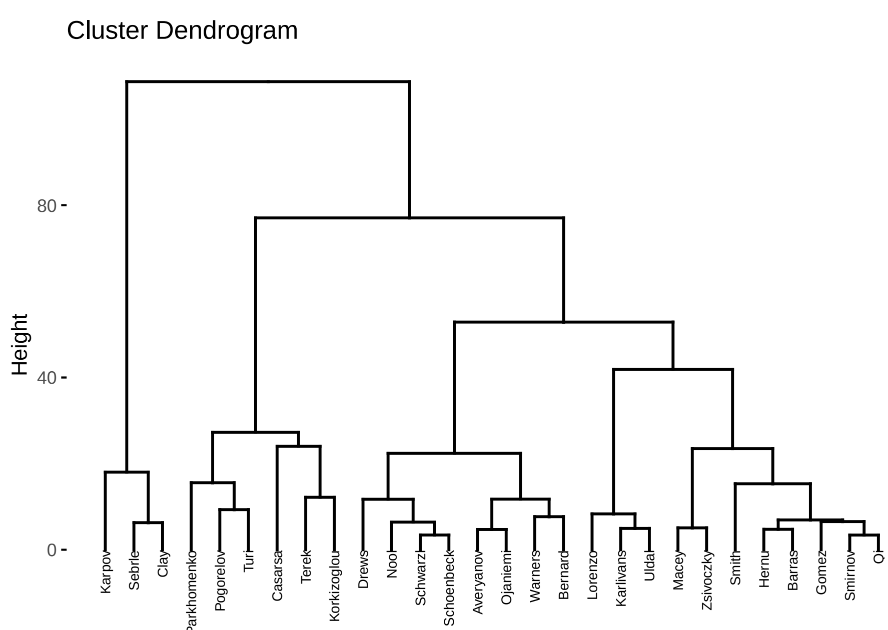


3.3 Classification par CAH

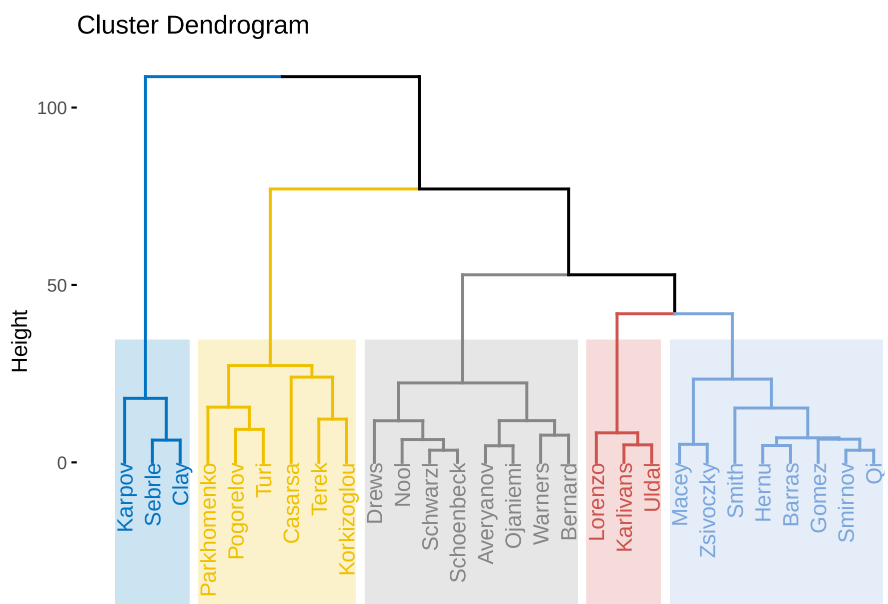
```
decathlon.cr <-  decathlon.active %>% scale(., scale=T, center=T)
decathlon.dist <- decathlon.cr %>% dist(., method = "euclidean")
decathlon.ward<-decathlon.dist^2 %>% hclust(., method = "ward.D")
```

3.3.1 Représentation graphique et choix du nombre de groupes

```
# Dendrogramme et choix du nombre de groupes
fviz_dend(decathlon.ward, cex = 0.5)
```



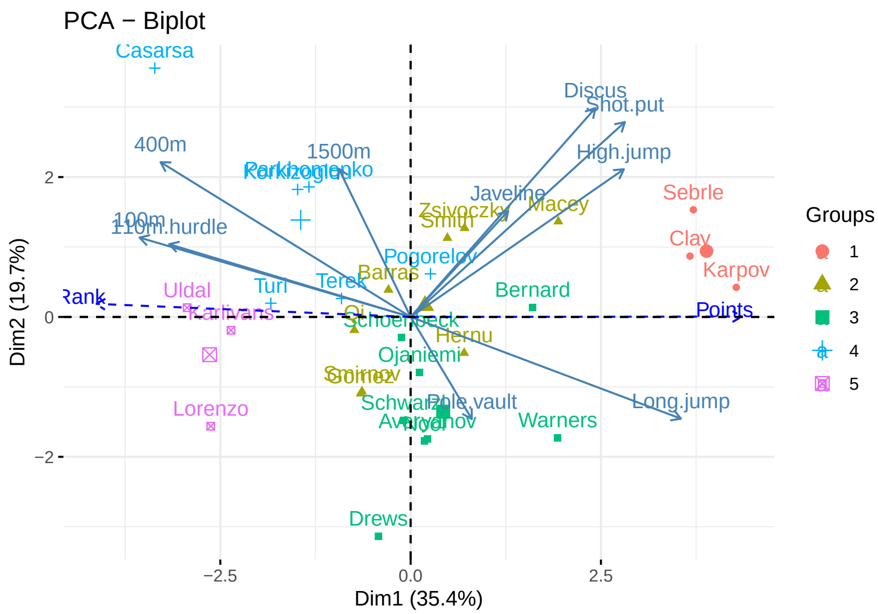
```r
fviz_dend(decathlon.ward,
          k=5,
          cex = 0.8,
          palette="jco",
          rect = TRUE, rect_fill = TRUE, # Rectangle autour des groupes
          rect_border = "jco",
          labels_track_height = 40
          )
```



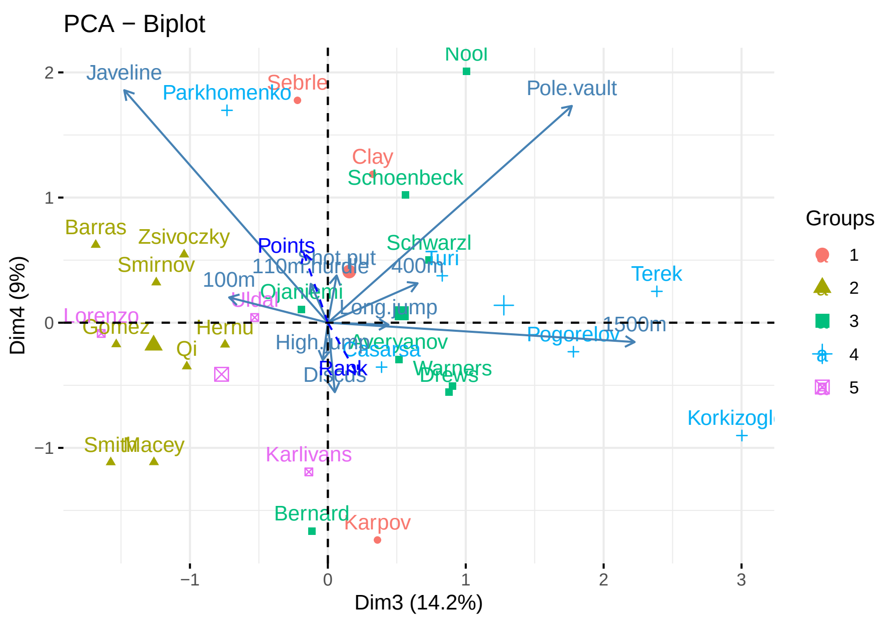

```r
cluster <- cutree(decathlon.ward, k =5)
```

3.3.2 Interprétation des groupes

```r
# visualiser les classes sur le premier plan factoriel de l'ACP
fviz_pca_biplot(res.pca,axes=c(1,2),habillage=as.factor(cluster))
```




```r
fviz_pca_biplot(res.pca,axes=c(3,4),habillage=as.factor(cluster))
```




```r
# moyennes des variables par groupe
knitr::kable(aggregate(decathlon.active, by=list(as.factor(cluster)),mean),digits=1)
```

Group.1	100m	Long.jump	Shot.put	High.jump	400m	110m.hurdle	Discus	Pole.vault	Javeline	1500m
1	10.6	7.9	15.8	2.1	48.1	14.1	50.2	4.8	65.3	280.0
2	11.0	7.2	14.7	2.0	49.1	14.5	45.1	4.5	61.0	268.1
3	10.8	7.5	14.3	1.9	49.2	14.4	42.2	4.9	56.8	275.5
4	11.1	6.9	14.9	2.0	51.2	14.8	44.4	4.8	56.8	293.2
5	11.2	7.1	13.3	1.9	50.3	15.2	42.2	4.5	57.1	274.5