Clustering

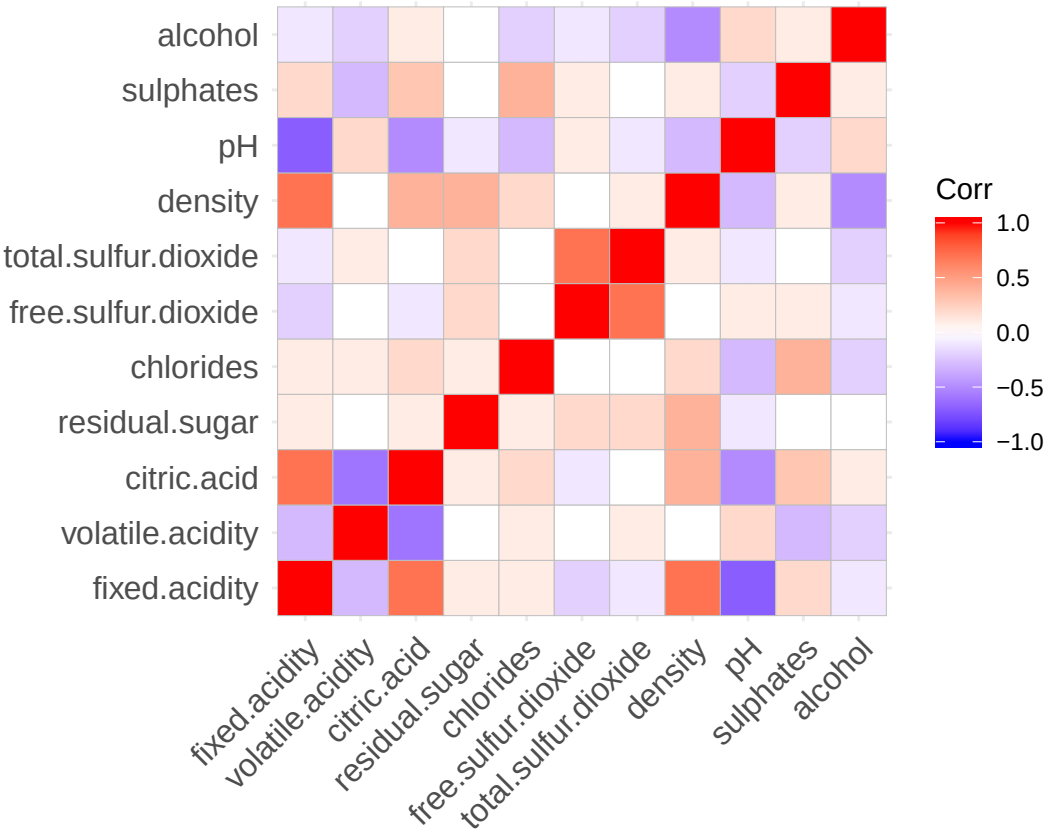
Andrea Huerfano

September 8, 2019

```
packages<-c("FactoMineR", "factoextra", "xtable", "NbClust", "plyr","GGally", "ggcorrplot")
lapply(packages, library, character.only=TRUE)
```

```
 winequality.red <- read.csv("C:/Users/Andrea/Desktop/wine/winequality-red.csv", sep=";")
winequality.red<-winequality.red[,-12]
```

```
corr <- round(cor(winequality.red), 1)
ggcorrplot(corr)
```




```
#colnames(winequality.red)
#winequality.red<-winequality.red[,-c(1,3,7)]
#corr <- round(cor(winequality.red), 1)
#ggcorrplot(corr)
```

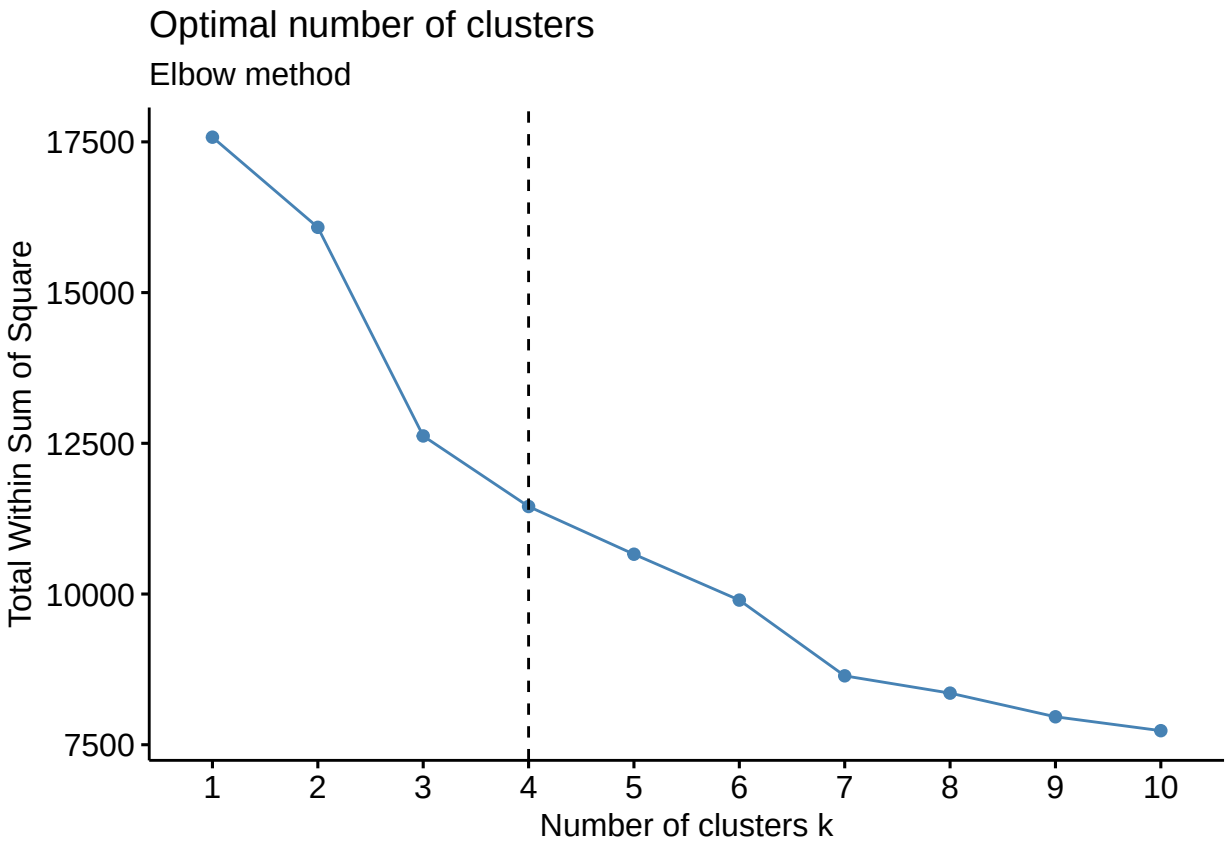
Se utiliza el paquete *NbClust* para calcular el número óptimo de grupos, en este caso a partir del indicador de *Hartigan*, el cuál sugiere el número óptimo de grupos y con este la mejor partición posible. El código asociado a este procedimiento se presenta a continuación

It can be concluded that standardization before clustering algorithm leads to obtain a better quality, efficient and accurate cluster result.

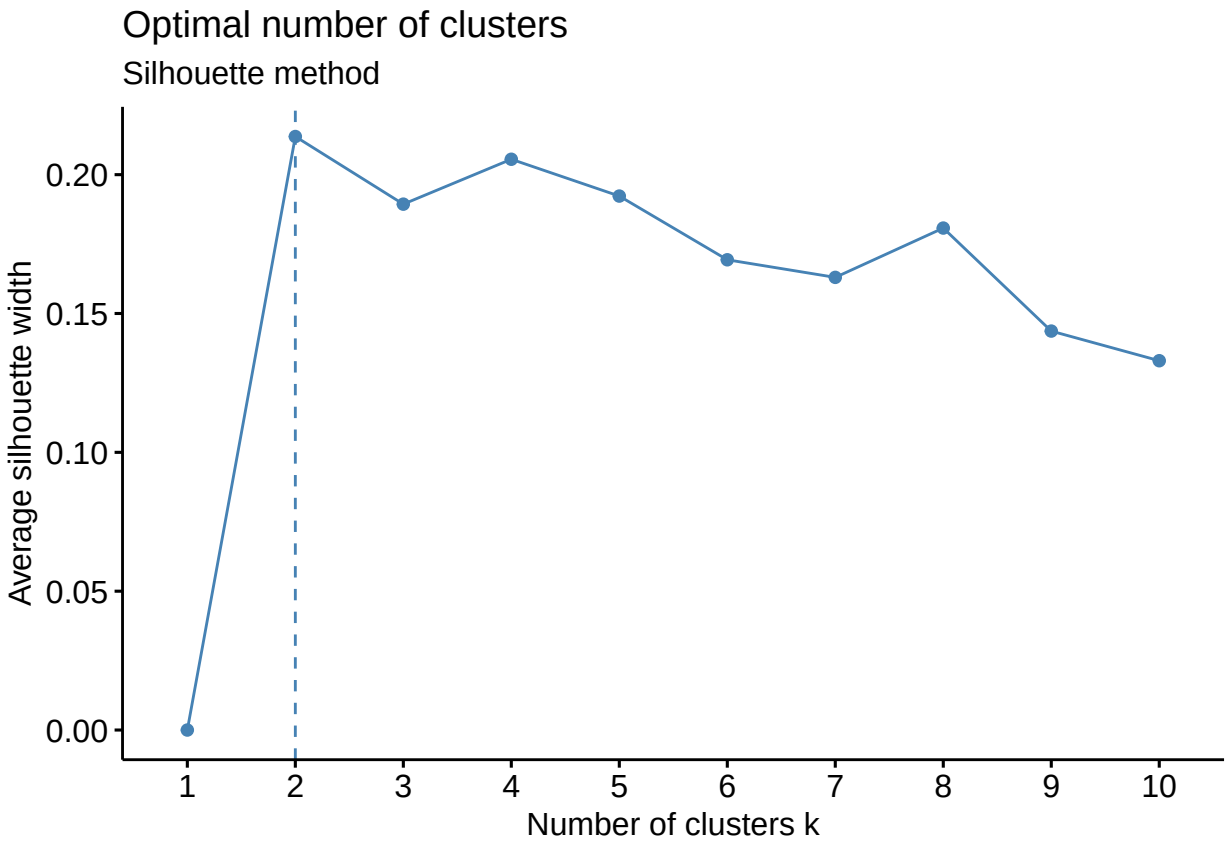
```
df<-scale(winequality.red)
res.wine<-NbClust(df, distance = "euclidean", min.nc = 2, max.nc = 10, method = "kmeans",
index = "hartigan")
res.wine$Best.nc
```

```
## Number_clusters     Value_Index
##          6.0000        255.1895
```

```
set.seed(21)
# Elbow method
elbow<-fviz_nbclust(df, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method"); elbow
```

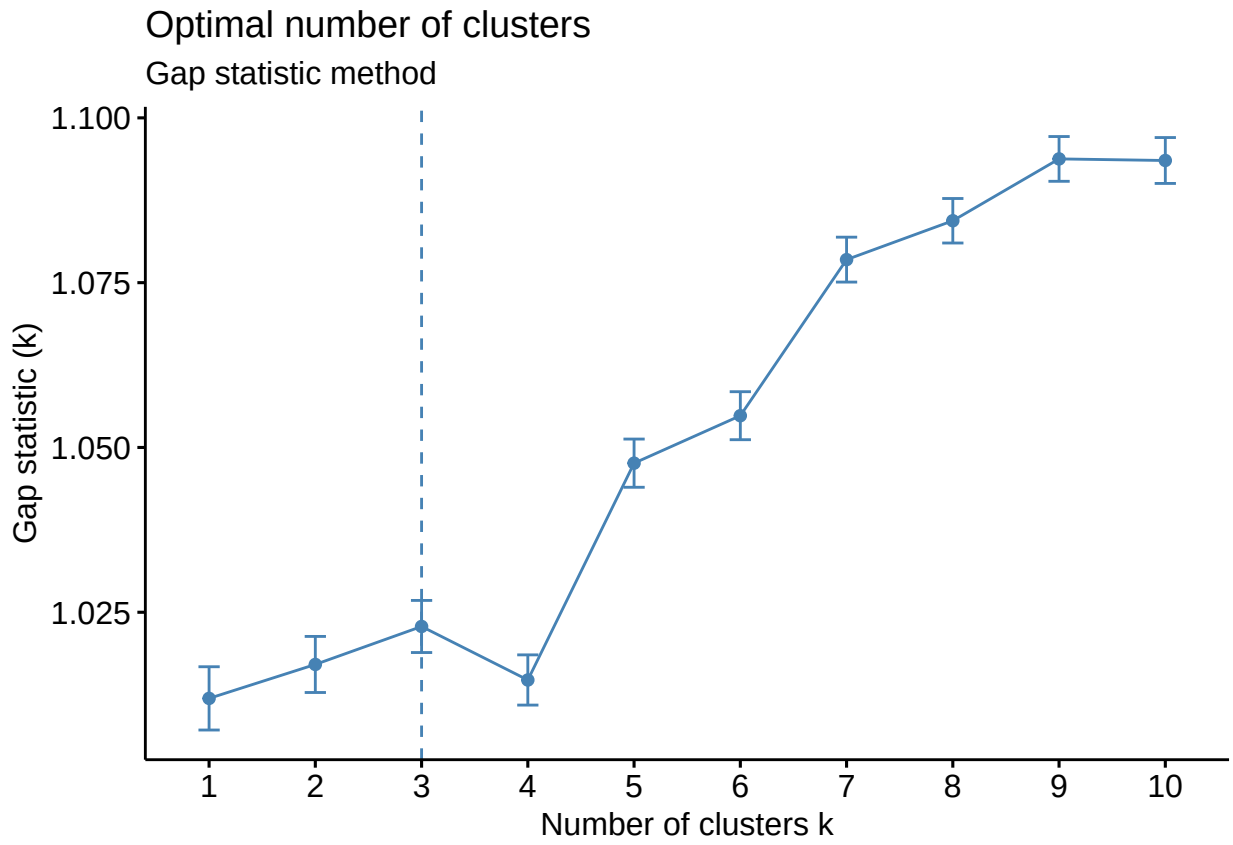



```
# Silhouette method
Silhouette<-fviz_nbclust(df, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method");Silhouette
```




```
# Gap statistic

gap<-fviz_nbclust(df, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+labs(subtitle = "Gap statis
```



Mean: 4 then number of cluster will be 4

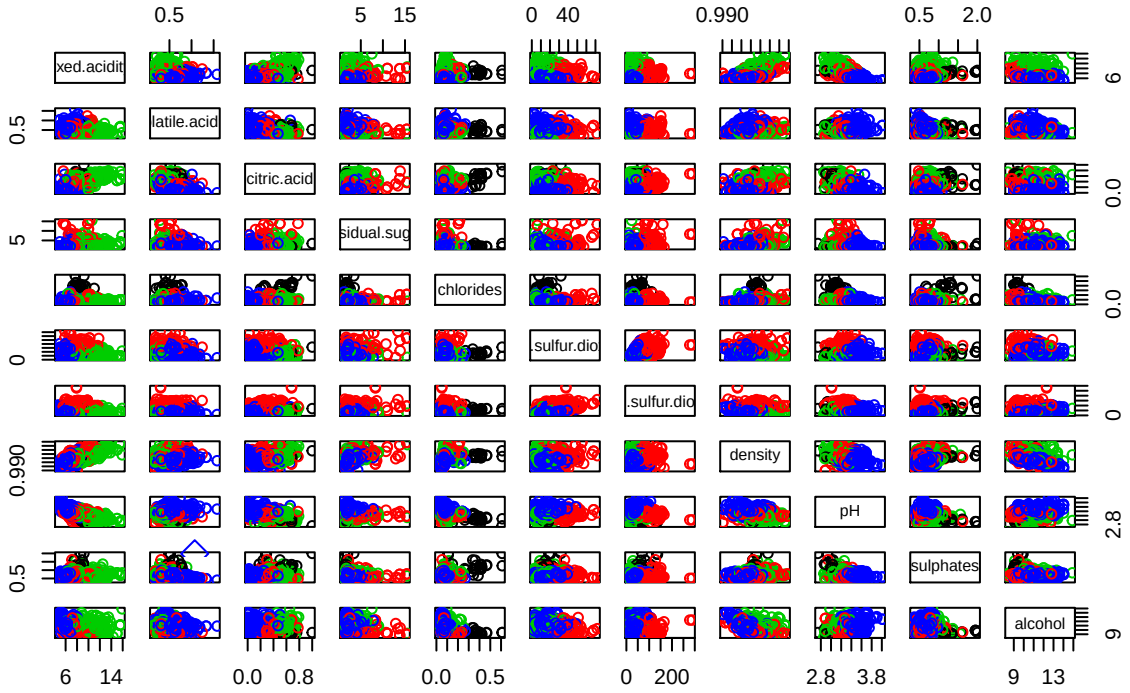
```
set.seed(10)
n_cluster<-4
kmedia2<-kmeans(df, n_cluster, nstart = 1000)
```

```
print(xtable(kmedia2$centers), comment=FALSE)
```

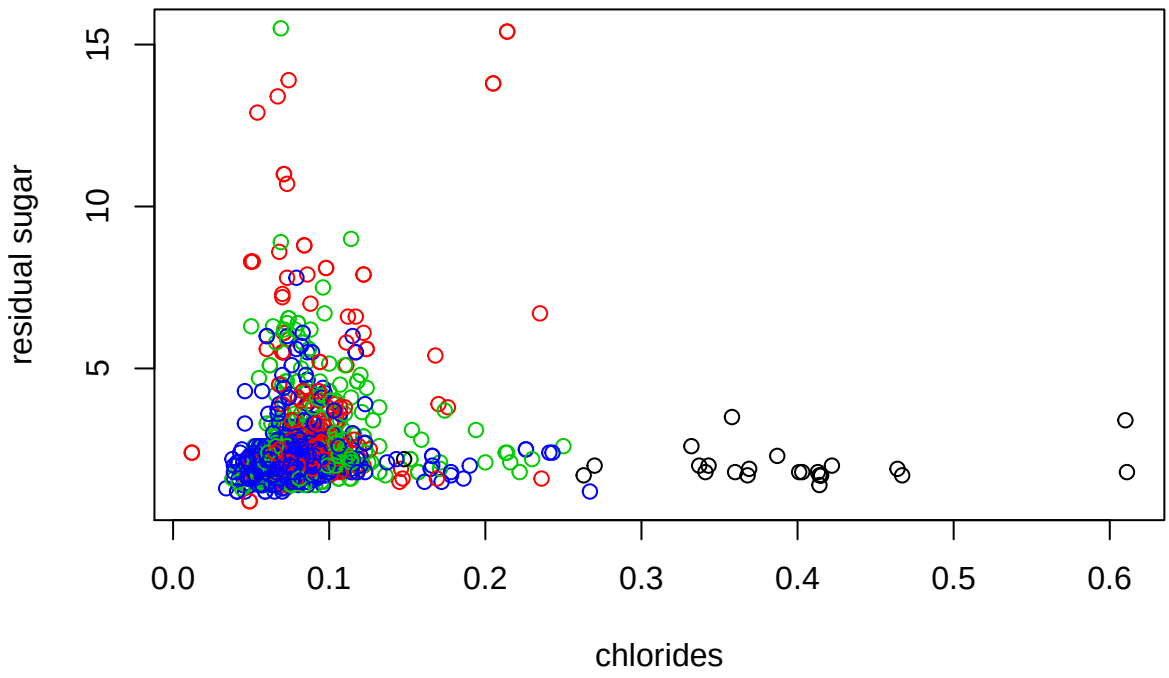
```
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrrrrrrr}
##   \hline
##  & fixed.acidity & volatile.acidity & citric.acid & residual.sugar & chlorides & free.sulfur.dioxide
##   \hline
## 1 & 0.10 & 0.00 & 1.18 & -0.39 & 5.78 & -0.05 & 0.51 & 0.18 & -1.74 & 3.66 & -0.87 \\
##   2 & -0.09 & 0.03 & 0.10 & 0.39 & -0.01 & 1.08 & 1.31 & 0.28 & -0.17 & -0.18 & -0.51 \\
##   3 & 1.06 & -0.73 & 1.02 & 0.06 & -0.06 & -0.51 & -0.53 & 0.45 & -0.69 & 0.36 & 0.37 \\
##   4 & -0.65 & 0.46 & -0.76 & -0.23 & -0.18 & -0.23 & -0.35 & -0.44 & 0.61 & -0.28 & 0.06 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
plot(winequality.red, col=kmedia2$cluster, main="Posición de los grupos respecto a las variables")
points(kmedia2$centers, col = kmedia2$cluster, pch = 23, cex = 1.5)
```

Posición de los grupos respecto a las variables




```
plot(winequality.red$chlorides, winequality.red$residual.sugar, col=kmedia2$cluster, xlab = "chlorides"
legend(0,100, legend=c('1', '2', '3', '4'),col=c("black","blue", "red","green" ),
pch = 18, cex = 1)
```




```
print(round(kmedia2$withinss,2))
```

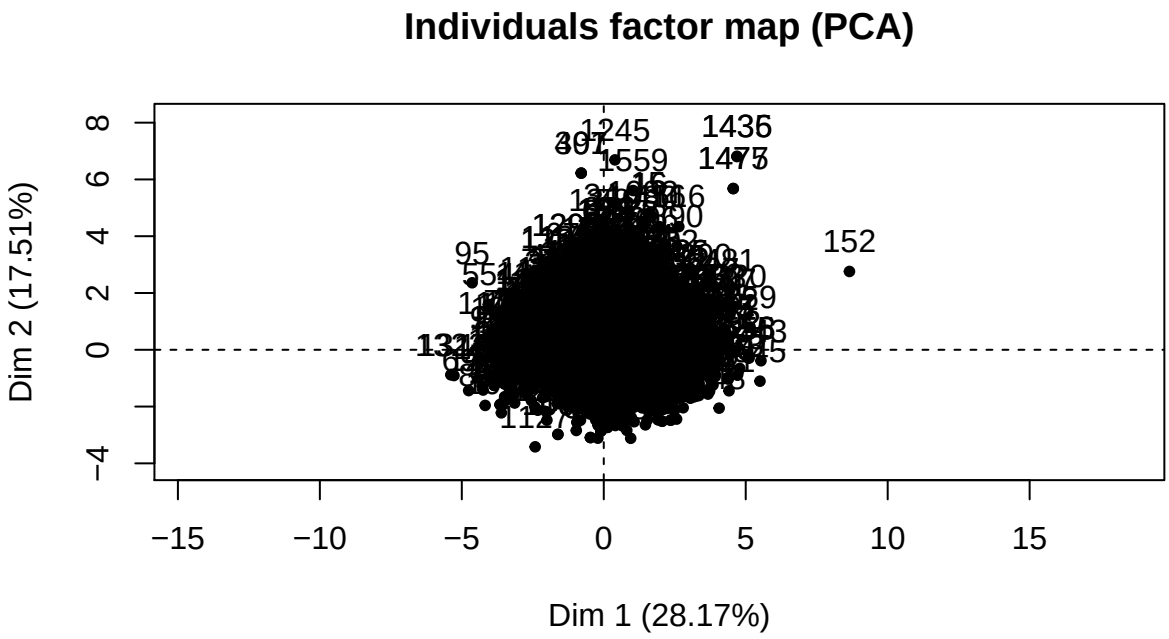
```
## [1]  428.05 3405.51 3246.43 4207.14
```

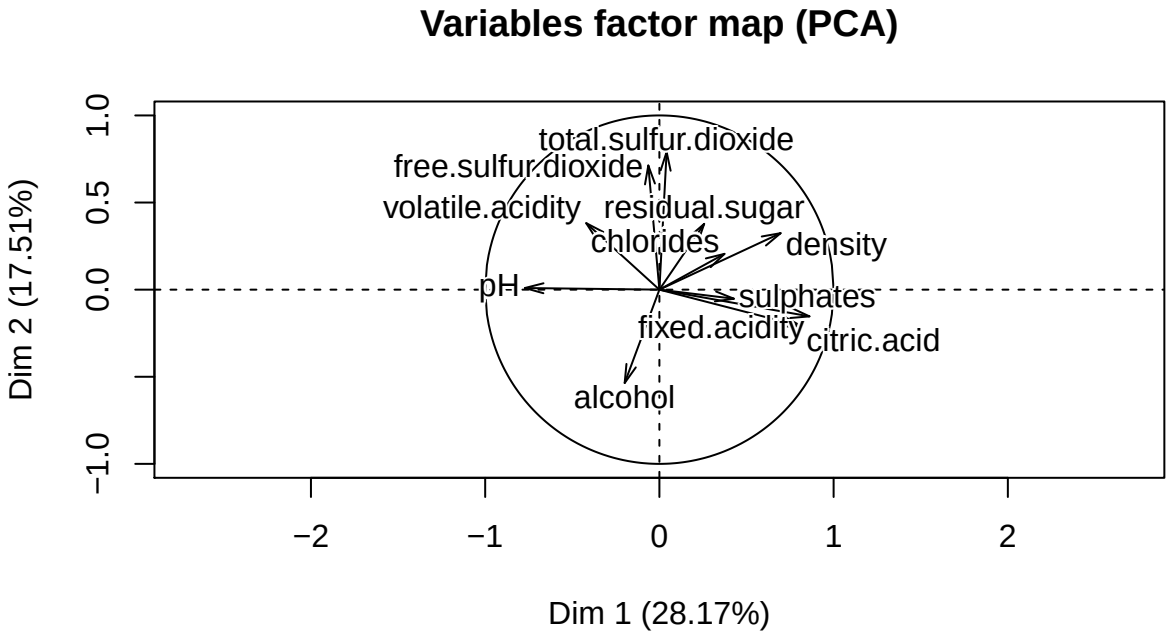
```
print(kmedia2$betweenss)
```

```
## [1] 6290.868
```

Nótese que la suma de cuadrados entre grupos es mayor que cada una de las sumas de cuadrados entre los grupos, lo que sugiere que los grupos entre si están bien diferenciados (alto grado de heterogeneidad) y en su interior presentan alto grado de homogeneidad.

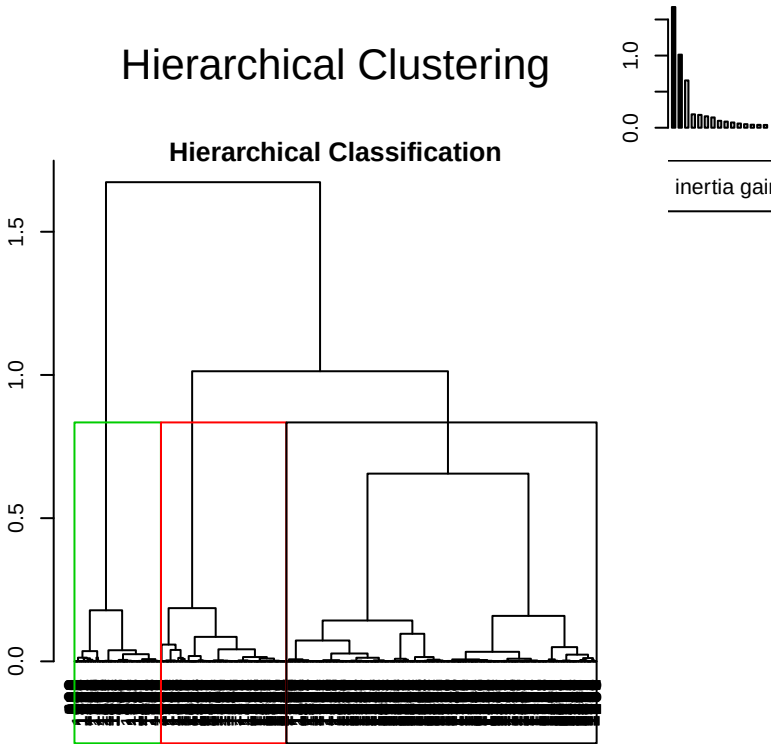
```
res.PCA<-PCA(df, ncp = 2)
```



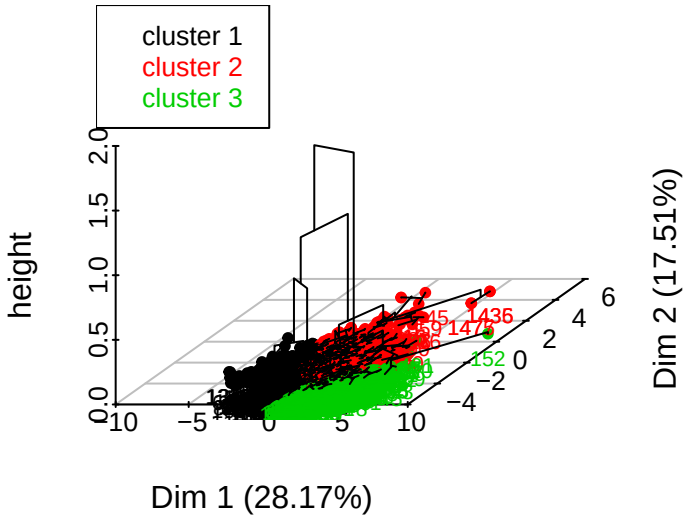


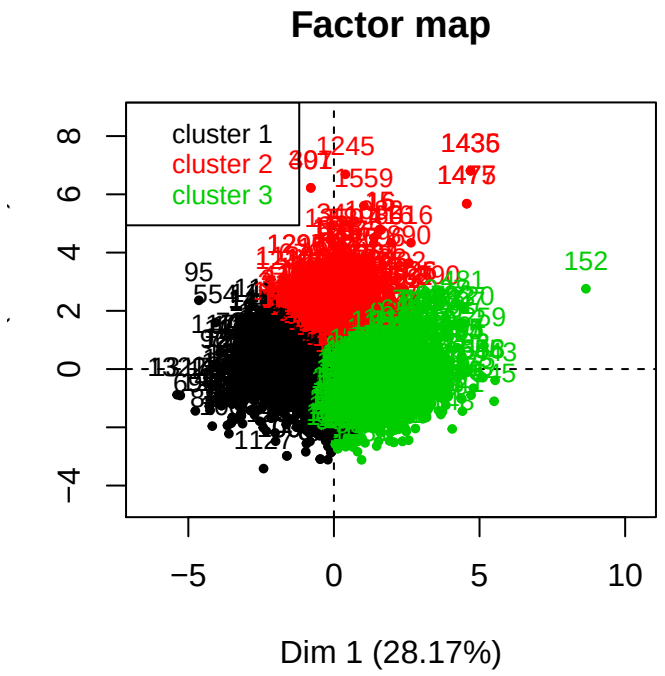
Ahora, utilicemos la función HPC sobre res.PCA

```
set.seed(10)
res.hcpc<-HCPC(res.PCA,  nb.clust = -1,min = 3, max = 6)
```



Hierarchical clustering on the factor map






```
#gmfd_kmeans(df, n.cl = 4, metric, p = NULL, k_trunc = NULL)
```