Stage symptômes viraux

Sébastien Guyader

December 14, 2018

Il s’agit de conduire une analyse en composantes principales (ACP) sur les données de symptômes viraux et d’indexation virale, réalisés sur 81 accessions d’igname.

## Import du fichier de données :

data <- read.csv("viro.csv", row.names = 1)

On vérifie le tableau importé :

head(data)

## Mosaique Chlorose Gaufrage Déformation\_foliaire Nanisme Enlacement  
## 16 0 2 0 0 0 0  
## 19 0 1 1 0 0 0  
## 22 3 3 1 1 0 2  
## 26 0 3 1 0 0 0  
## 34 2 2 1 0 0 0  
## 35 1 2 2 1 0 2  
## Points\_verts YMV YMMV Potex Ampelo Maclura Séco Badna  
## 16 0 0 1 0 1 0 0 0  
## 19 0 0 1 0 1 0 0 0  
## 22 0 0 1 0 1 1 0 1  
## 26 1 0 1 0 1 0 1 0  
## 34 0 0 1 0 1 0 1 0  
## 35 0 0 0 0 1 0 0 0

#data$Points\_verts <- NULL

## ACP

On va charger les librairies nécessaires : FactoMineR et factoextra :

library(FactoMineR) ; library(factoextra)

## Loading required package: ggplot2

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ tibble 1.4.2 ✔ purrr 0.2.5  
## ✔ tidyr 0.8.2 ✔ dplyr 0.7.8  
## ✔ readr 1.2.0 ✔ stringr 1.3.1  
## ✔ tibble 1.4.2 ✔ forcats 0.3.0

## ── Conflicts ───────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

On commence par faire l’analyse en composantes principales (PCA) :

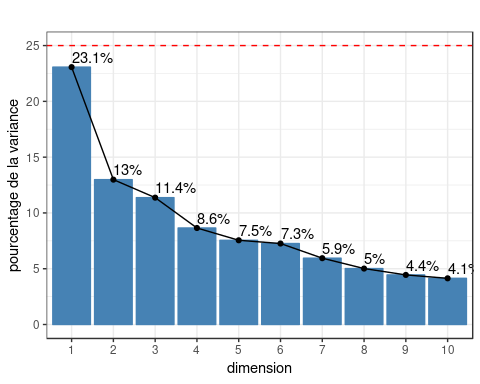
res.pca <- PCA(data, graph = F, ncp=5)

On détermine valeurs propres des axes :

1/(nrow(data)-1)\*100

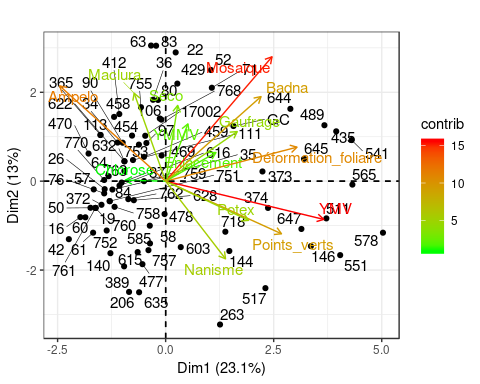
## [1] 1.25

val\_propre <- 1/((5)-1)  
  
fviz\_screeplot(res.pca, addlabels=TRUE) +  
 geom\_hline(yintercept = val\_propre\*100, linetype = 2, color = "red") +  
 xlab("dimension") + ylab("pourcentage de la variance") +  
 ggtitle("") +  
 theme\_bw()



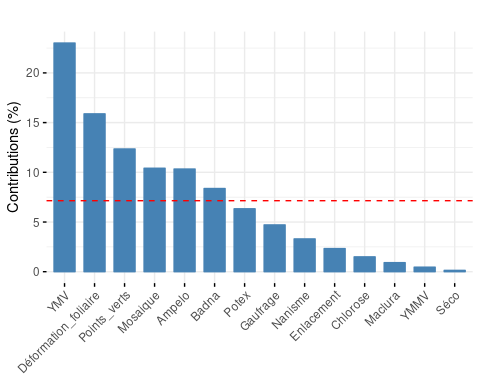
Graphique des variables, avec coloration en fonction de leur contribution :

fviz\_pca\_biplot(res.pca, col.var="contrib", repel=T) +  
 scale\_color\_gradient(low="green", high="red") +  
 ggtitle("") +  
 theme\_bw()



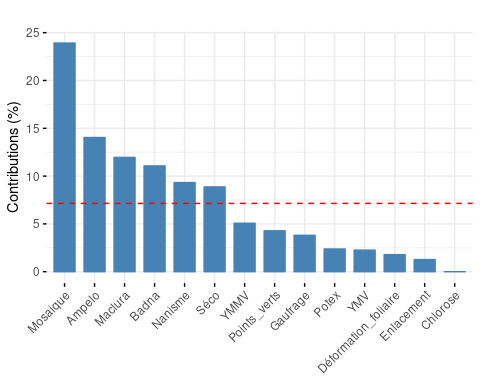
Contributions des variables sur l’axe 1 :

fviz\_contrib(res.pca, choice="var", axes = 1) +  
 ggtitle("")



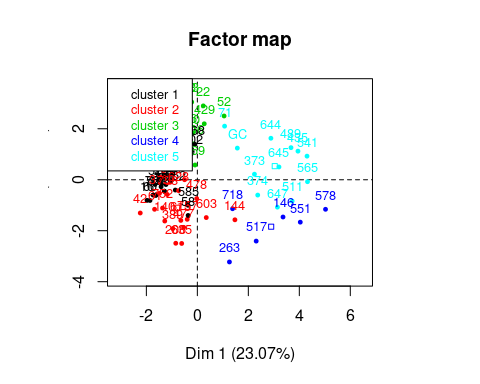
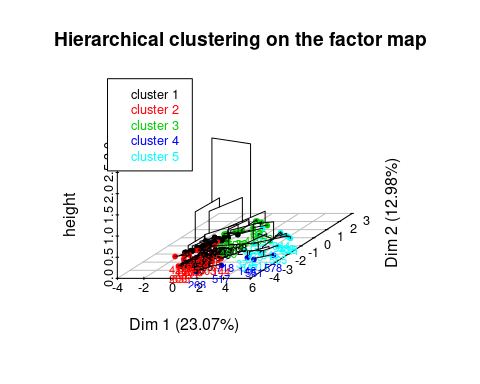
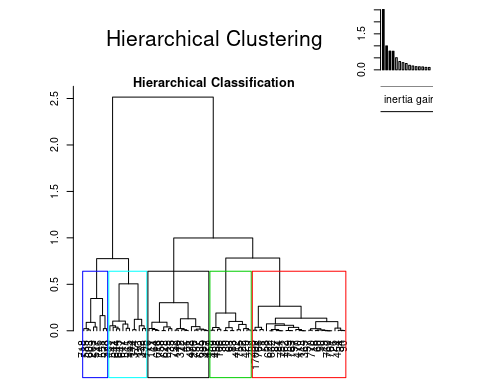
Contributions des variables sur l’axe 2 :

fviz\_contrib(res.pca, choice="var", axes = 2) +  
 ggtitle("")

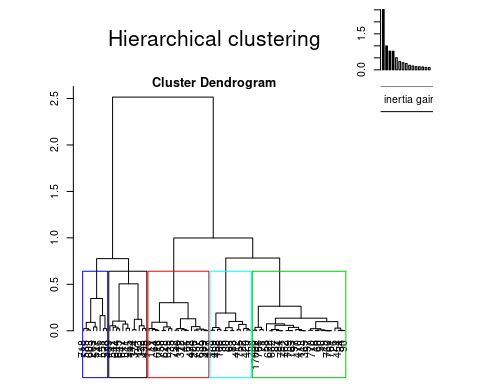


## Classification hiérarchique

res.hcpc <- HCPC(res.pca, nb.clust=-1, cluster.CA="rows")



plot(res.hcpc, choice="tree", axes=c(1,2))



clust.plot <- res.hcpc %>% fviz\_cluster(repel=T, show.clust.cent=F, axes=c(1,2),  
 ellipse.type = "norm", legend="none",  
 main="Analyse Factorielle des Correspondances",  
 ggtheme=theme\_bw())  
  
CA.cols <- as.data.frame(res.pca$col)  
  
library(ggpubr)

## Loading required package: magrittr

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

clust.plot +  
 geom\_exec(geom\_point, data=CA.cols, x="coord.Dim.1", y="coord.Dim.2", shape=1:5, size=3, color="red") +  
 geom\_hline(yintercept = 0, linetype = 2, color = "black") +  
 geom\_vline(xintercept = 0, linetype = 2, color = "black")

