

```
# Rotation of the results of a PCA: the spider data

spiders=read.table(file.choose())      # Fichier "Spiders_28x12_spe.txt"
# Transformation de Hellinger des abondances d'araignées
spiders.hel = decostand(spiders, "hellinger")

# ACP avec la fonction prcomp() de {stats}
pca.spiders = prcomp(spiders.hel)
# Extraire le fichier de position des sites sur les 3 premiers axes ACP
spiders.sites = summary(pca.spiders)$x[,1:3]

library(rgl)
rgl.open()                             # Ouvrir la fenêtre graphique de "rgl"
rgl.points(spiders.sites,color="green",size=6)

axis.ranges = apply(spiders.sites,2,range) # min et max de chaque axe
axis.ranges
#           Comp.1      Comp.2      Comp.3
# [1,] -0.5032444 -0.4630804 -0.3999831
# [2,]  0.8617815  0.4964801  0.4714523

rgl.lines(x=c(-1,1),y=c(0,0),z=c(0,0))
rgl.lines(y=c(-1,1),x=c(0,0),z=c(0,0))
rgl.lines(z=c(-1,1),x=c(0,0),y=c(0,0))

rgl.texts(1,0,0,"Axis1")
rgl.texts(0,1,0,"Axis2")
rgl.texts(0,0,1,"Axis3")

-----

# PCA biplot from prcomp {stats}
biplot(pca.spiders)

# PCA biplots from rda {vegan}
library(vegan)
spiders.pca = rda(spiders.hel)
par(mfrow=c(1,2))
biplot(spiders.pca, scaling=1)
biplot(spiders.pca, scaling=2)

=====

# How many groups of sites are there in file 'spiders' ?

# Ward agglomerative clustering

spiders.hel = decostand(spiders, "hel")
spiders.hel.D = dist(spiders.hel)

spiders.clust.ward = hclust(spiders.hel.D^2, method="ward")
spiders.clust.ward$height = sqrt(spiders.clust.ward$height)
plot(spiders.clust.ward, hang=-1)

=====

# Partitioning by cascadeKM()
```

```
spiders=read.table(file.choose()) # Fichier 'Spiders_28x12 with sp names.txt'
spiders.hel = decostand(spiders, "hellinger")

spiders.km.CH = cascadeKM(spiders.hel, 2, 10, iter = 100)
spiders.km.CH$results
spiders.km.CH$size
toto <- plot(spiders.km.CH)

cross.out = cross.cascadeKM(spiders.hel, spiders.km.CH, 2, 10)
cross.out$AIC
# Quel nombre de groupes obtient la plus petite erreur de validation croisée
# (statistique "cvre") ?
spiders.km.CH$partition[,1:6]

spiders.km.ssi = cascadeKM(spiders.hel, 2, 10, criterion="ssi")
spiders.km.ssi$results
spiders.km.ssi$size
toto <- plot(spiders.km.ssi)
spiders.km.ssi$partition[,1:6]

=====
=====

# Rotation of 3-D data, 6 points (from file "PCA_CA_and_PCoA_algebra")

library(rgl)

# Example PCA, 6 points, 3 variables
mat <- matrix(c(2,3,5,7,9,10,1,4,0,6,2,5,0,1,-1,-1,1,0),6,3)

rgl.open()
rgl.points(mat,color="green",size=6)

limits = apply(mat,2,range)
limits

rgl.lines(x=c(-10,10),y=c(0,0),z=c(0,0))
rgl.lines(y=c(-6,6),x=c(0,0),z=c(0,0))
rgl.lines(z=c(-2,2),x=c(0,0),y=c(0,0))

rgl.texts(limits[2,1],0,0,"Var1")
rgl.texts(0,limits[2,2],0,"Var2")
rgl.texts(0,0,2,"Var3")

=====

# Rotation of random data points

library(rgl)

vec1=matrix(rnorm(100,0,1),100,1)
vec2=matrix(rnorm(100,0,3),100,1)
vec3=matrix(rnorm(100,0,6),100,1)
mat=cbind(vec1,vec2,vec3)

rgl.open()
```

```
rgl.points(mat,color="green",size=6)

limits = apply(mat,2,range)
limits

rgl.lines(x=c(limits[1,1],limits[2,1]),y=c(0,0),z=c(0,0))
rgl.lines(y=c(limits[1,2],limits[2,2]),x=c(0,0),z=c(0,0))
rgl.lines(z=c(limits[1,3],limits[2,3]),x=c(0,0),y=c(0,0))

rgl.texts(limits[2,1],0,0,"Var1")
rgl.texts(0,limits[2,2],0,"Var2")
rgl.texts(0,0,limits[2,3],"Var3")

=====

# Rotation of points with an outlier

# Example PCA, 6 points, 3 variables, with one outlier
mat.out <- matrix(c(2,3,5,7,9,10,1,4,0,6,2,50,0,1,-1,-1,1,0),6,3)

library(rgl)
rgl.open()
rgl.points(mat.out,color="green",size=6)

# limits = apply(mat.out,2,range)
# limits

rgl.lines(x=c(-10,10),y=c(0,0),z=c(0,0))
rgl.lines(y=c(-10,50),x=c(0,0),z=c(0,0))
rgl.lines(z=c(-2,2),x=c(0,0),y=c(0,0))

rgl.texts(10,0,0,"Var1")
rgl.texts(0,50,0,"Var2")
rgl.texts(0,0,2,"Var3")

=====
=====
```