```
library(fda)
library(cluster)      #the 4 libraries used for data analysis and display
library(evd)
library(rgl)


########################################
###########     Figure 3:     ###########
########################################

### Example of three kernels estimates(grey bolded curves) of a centered
Gaussian
### density(black bolded curve). The roughness of the estimates is
controlled by
### a smoothing parameter (bw). Each density is estimated on the same
dataset
### composed of 50 values of x randomly drawn from the Gaussian
distribution.



xabs=seq(-4,4,0.1)
n=length(xabs)
xmin=min(xabs)
xmax=max(xabs)
x=dnorm(xabs)
xx=rnorm(50)
xden1=density(xx,bw=0.3,from=xmin,to=xmax,n=n)
xden2=density(xx,bw=0.5,from=xmin,to=xmax,n=n)
xden3=density(xx,bw=1,from=xmin,to=xmax,n=n)
xest=cbind(x,xden1$y,xden2$y,xden3$y)
matplot(xabs,xest,col=c(1,rep("dark
grey",3)),lwd=c(2,1,1,1),type="l",lty=1,xlab="x",ylab="density")
rug(xx)
X11()



########################################
###########     Figure 4:     ###########
########################################

### The six families of curves including the reference density for each
class
### (black bolded curves) and a sample of kernel density estimate. Each
random
### function in class Cj is estimated using 50 points randomly drawn from
the
### reference density fj.


gamma=3/5

dgumbel_inv=function(x,loc,sca){
 z=exp(-(-x-loc)/sca)
 res=1/sca*z*exp(-z)
 return(res)
 }
par(mfcol=c(2,3))
#mar=c(2,1,1,1)
xabs=seq(-6,6,0.1)
```

```
xmin=min(xabs)
xmax=max(xabs)
n=length(xabs)
f1=dnorm(xabs,0,1.5)
f2=dnorm(xabs,0,3)
fa=dnorm(xabs,-1,0.5)
fb=dnorm(xabs,1,0.5)
f3=gamma*fa+(1-gamma)*fb
f4=gamma*fb+(1-gamma)*fa
f5=dgumbel(xabs,loc=-3,sca=1)
f6=dgumbel_inv(xabs,loc=-3,sca=1)
########################################
##f1
xden=matrix(-99,n,10)
for(k in 1:10){
  x=rnorm(50,0,1.5)
  xden[,k]=density(x,from=xmin,to=xmax,n=n)$y
  }

matplot(xabs,xden,type="l",lty=1,xlab="x",ylab="density
f1",ylim=c(0,0.5),col="dark grey")
lines(xabs,f1,col=1,lwd=2)
abline(v=0,lty=2)
text(-5,0.4,"C1",cex=2)
########################################
##f2
xden=matrix(-99,n,10)
for(k in 1:10){
  x=rnorm(50,0,3)
  xden[,k]=density(x,from=xmin,to=xmax,n=n)$y
  }
matplot(xabs,xden,type="l",lty=1,xlab="x",ylab="density
f2",ylim=c(0,0.5),col="dark grey")
lines(xabs,f2,col=1,lwd=2)
abline(v=0,lty=2)
text(-5,0.4,"C2",cex=2)
########################################
##f3
xden=matrix(-99,n,10)
for(k in 1:10){
  xa=rnorm(3/5*50,-1,1/2)
  xb=rnorm(2/5*50,1,1/2)
  x=c(xa,xb)
  xden[,k]=density(x,from=xmin,to=xmax,n=n)$y
  }
matplot(xabs,xden,type="l",lty=1,xlab="x",ylab="density
f3",ylim=c(0,0.5),col="dark grey")
lines(xabs,f3,col=1,lwd=2)
abline(v=0,lty=2)
text(-5,0.4,"C3",cex=2)
########################################
##f4
xden=matrix(-99,n,10)
for(k in 1:10){
  xa=rnorm(2/5*50,-1,1/2)
  xb=rnorm(3/5*50,1,1/2)
  x=c(xa,xb)
  xden[,k]=density(x,from=xmin,to=xmax,n=n)$y
  }
matplot(xabs,xden,type="l",lty=1,xlab="x",ylab="density
f4",ylim=c(0,0.5),col="dark grey")
```

```
lines(xabs,f4,col=1,lwd=2)
abline(v=0,lty=2)
text(-5,0.4,"C4",cex=2)
#########################################
##f5
xden=matrix(-99,n,10)
for(k in 1:10){
  x=rgumbel(50,loc=-3,sca=1)
  xden[,k]=density(x,from=xmin,to=xmax,n=n)$y
  }
matplot(xabs,xden,type="l",lty=1,xlab="x",ylab="density
f5",ylim=c(0,0.5),col="dark grey")
lines(xabs,f5,col=1,lwd=2)
abline(v=0,lty=2)
text(-5,0.4,"C5",cex=2)
#########################################
##f6
xden=matrix(-99,n,10)
for(k in 1:10){
  x=-rgumbel(50,loc=-3,sca=1)
  xden[,k]=density(x,from=xmin,to=xmax,n=n)$y
  }
matplot(xabs,xden,type="l",lty=1,xlab="x",ylab="density
f6",ylim=c(0,0.5),col="dark grey")
lines(xabs,f6,col=1,lwd=2)
abline(v=0,lty=2)
text(-5,0.4,"C6",cex=2)
X11()


#########################################
###########     Figure 5:     ###########
#########################################

### The research of the optimal number of clusters. First, the invariance
by
### orientation is not applied to compute the distance matrix. Then, the
### invariance by symmetry is taken into account for the computing.


s=100
nobs=100
n=600
ind<-c(rep(1,100),rep(2,100),rep(3,100),rep(4,100),rep(5,100),rep(6,100))
ind2<-c(rep(1,100),rep(1,100),rep(2,100),rep(3,100),rep(4,100),rep(4,100))

  yy = list()
  for(i in 1:n){
  if(ind[i]==1) yy[[i]] = c(rnorm(3*s/5,-1,0.5),rnorm(2*s/5,1,0.5))
  if(ind[i]==2) yy[[i]] = c(rnorm(2*s/5,-1,0.5),rnorm(3*s/5,1,0.5))
  if(ind[i]==3) yy[[i]] = rnorm(s,0,1.5)
  if(ind[i]==4) yy[[i]] = rnorm(s,0,3)
  if(ind[i]==5) yy[[i]] = rgumbel(s,-3,1)
  if(ind[i]==6) yy[[i]] = rgumbel(s,-3,1)
  }

dens<-list()
xabs<-seq(-7,7,length=nobs)
  for(i in 1:n){
  dens[[i]]<-density(yy[[i]],from=-7,to=7,n=nobs,bw=bw.ucv(yy[[i]]))$y
  }
```

```
yt=xt=vector()
  for ( i in seq(1,length(dens))){
  xx1  <- seq(1,length(dens[[i]]),len=201)
  s1<-smooth.spline(dens[[i]])
  xt<-cbind(xt,predict(s1,xx1)$y)
  yt<-cbind(yt,xx1)
  }

basis=create.fourier.basis(c(0,1),21)
pop=data2fd(xt,basisobj=basis)
coefnorms=pop$coefs/matrix(rep(pop$coefs[1,],21),nrow=21,byrow=TRUE)
coefnorms[,ind==6]=coefnorms[,ind==6]*c(rep(c(1,-1),10),1)
popnorm=fd(coefnorms,basis)

par()
plot(popnorm,col=ind,xlab="support normalisé x",ylab="f(x)")

#functional acp
aa=pca.fd(popnorm,3)
plot3d(aa$scores[,1],aa$scores[,2],aa$scores[,3],col=ind,size=2,axes=F)
par(mfrow=c(1,3))
plot(aa$scores[,1],aa$scores[,2],col=ind)
plot(aa$scores[,2],aa$scores[,3],col=ind)
plot(aa$scores[,1],aa$scores[,3],col=ind)


#optimal number of groups with Silhouette plot
res=vector()
  for (k in 2:12) {
  kme=fanny(aa$scores,k)
  res[k]=summary(silhouette(kme))[[1]][4]
  }
par()
plot(2:12,res[-1],type="l", xlab="Number of groups", ylab="Silhouette
coefficient")
points(2:12,res[-1], pch=10, col="red", cex=2)
axis(side=1, 2:12, labels=FALSE)
title("Silhouette plot: Optimal number of groups without invariance by
orientation", font.main=3, adj=1)
X11()

#acp with distance invariant to orientation
dis=dist_inv2(coefnorms,21)
resmds=cmdscale(dis,3,eig=T)
afm=resmds$points
plot3d(afm[,1],afm[,2],afm[,3],col=ind,size=2,axes=F)

par(mfrow=c(1,3))
plot(afm[,1],afm[,2],col=ind)
plot(afm[,2],afm[,3],col=ind)
plot(afm[,1],afm[,3],col=ind)
X11()
#optimal number of groups with Silhouette plot
res=vector()
  for (k in 2:12) {
  kme=fanny(afm,k)
  res[k]=summary(silhouette(kme))[[1]][4]
  }

par()
```

```
plot(2:12,res[-1],type="l", xlab="Number of groups", ylab="Silhouette
coefficient")
points(2:12,res[-1], pch=10, col="red", cex=2)
axis(side=1, 2:12, labels=FALSE)
title("Silhouette plot: Optimal number of groups with invariance by
orientation", font.main=3, adj=1)


#test bagging
a=bagging(afm,ind2,4,30)
mean(a)

#########################################
###########     Figure 6:     ###########
#########################################

### The first alteration experiment:  the alteration of the relative
abundance
### of class C1 to C6.

resultot=resultot3=vector()
resultot2=list()
tester=seq(from=0.05,to=0.165,length=50)
tester=tester[50:1]
pb <- winProgressBar(title = "progress bar", min = 0,max = 50, width = 300)
for(k in 1:50){
Sys.sleep(0.1)
setWinProgressBar(pb, k, title=paste( round(k/50*100, 0),"% done"))
nobs=600
n=100
s=100
spa=round(tester[k]*n)
spb=n-5*spa
ind<-c(rep(1,spa),rep(2,spa),rep(3,spb),rep(4,spa),rep(5,spa),rep(6,spa))
ind2<-c(rep(1,spa),rep(1,spa),rep(2,spb),rep(3,spa),rep(4,spa),rep(4,spa))
yy = list()
               for(i in 1:n){
               if(ind[i]==1) yy[[i]] = c(rnorm(3*s/5,-
1,0.5),rnorm(2*s/5,1,0.5))
               if(ind[i]==2) yy[[i]] = c(rnorm(2*s/5,-
1,0.5),rnorm(3*s/5,1,0.5))
               if(ind[i]==3) yy[[i]] = rnorm(s,0,1.5)
               if(ind[i]==4) yy[[i]] = rnorm(s,0,3)
    if(ind[i]==5) yy[[i]] = rgumbel(s,-3,1)
    if(ind[i]==6) yy[[i]] = rgumbel(s,-3,1)
    }
dens<-list()
xabs<-seq(-7,7,length=nobs)
               for(i in 1:n){
               dens[[i]]<-density(yy[[i]],from=-
7,to=7,n=nobs,bw=bw.nrd0(yy[[i]]))$y
               }
yt=xt=vector()
    for ( i in seq(1,length(dens))){
    xx1  <- seq(1,length(dens[[i]]),len=201)
    s1<-smooth.spline(dens[[i]])
    xt<-cbind(xt,predict(s1,xx1)$y)
    yt<-cbind(yt,xx1)
    }
basis=create.fourier.basis(c(0,1),21)
pop=data2fd(xt,basisobj=basis)
```

```
coefnorms=pop$coefs/matrix(rep(pop$coefs[1,],21),nrow=21,byrow=TRUE)
coefnorms[,ind==6]=coefnorms[,ind==6]*c(rep(c(1,-1),10),1)
dis=dist_inv2(coefnorms,21)
resmds=cmdscale(dis,3,eig=T)
afm=resmds$points
newpar <- par()
par(mfrow=c(1,3))
plot(afm[,1],afm[,2],col=ind2)
plot(afm[,2],afm[,3],col=ind2)
plot(afm[,1],afm[,3],col=ind2)
par(newpar)

#clusplot(pam(afm,4,metric="euclidean")
#calcul of the optimal number of groups
#res2=vector()
# for (f in 2:10) {
# kme=pam(afm,f)
# res2[f]=summary(silhouette(kme))[[1]][4]
# }

tab=afm
flag=ind2
nbgroup=4 #which.max(res2) if we would like to consider the optimal number
of group
nbtir=200
nba1=round(2*(dim(tab)[1])/3)
sel=sort(sample(1:dim(tab)[1],nba1,replace=F))
sel2=setdiff(seq(1,dim(tab)[1]),sel)
a1= tab[sel,]
deucl<-function(x,y) {sqrt(sum((x-y)^2))}
txtot=vector()
  for (d in seq(1,nbtir)){
  tirage=sort(sample(1:nba1,nba1,replace=TRUE))
  kme=pam(daisy(a1[tirage,]),nbgroup, diss =TRUE)
  gp=by(tab[sel[tirage],],kme$cluster,list)
  gpm=lapply(gp,mean)
  gpmtab= t(matrix(unlist(gpm),ncol=nbgroup))
  compare=flag[sel[tirage]]
  tx=by(compare,kme$cluster,list)
  txtab=lapply(tx,table)
  txgrp=lapply(txtab,which.max)
  txgp=as.numeric(lapply(txgrp,names))
  la2= length(sel2)
  a2=tab[sel2,]
  test=flag[sel2]
  res=vector()
    for (n in seq(1,la2)){
    qmin=vector()
    qmin=apply(gpmtab,1,y=a2[n,],deucl)
    res=c(res,is.element(test[n],unlist(txgp)[which.min(qmin)]) )
    }
  tx1=mean(res[flag[sel2]==1])
  tx2=mean(res[flag[sel2]==2])
  tx3=mean(res[flag[sel2]==3])
  tx4=mean(res[flag[sel2]==4])
  txtot=cbind(txtot,c(tx1,tx2,tx3,tx4))}
  resultot=c(resultot,mean(txtot))
  resultot3=c(resultot3,sd(txtot))
  resultot2[[k]]=txtot
}
close(pb)
```

```
label=cbind(tester,tester,tester,tester,tester,(1-5*tester))
rownames(label)=  1:50
colnames(label)=c("C1","C2","C3","C4","C5","C6")
mp=barplot(t(label),legend= colnames(label), xlim = c(0, 70))
points(mp,resultot,pch=20, col=1, cex=2)


#########################################
###########     Figure 9:     ###########
#########################################

# The second sample alteration experiment: low noise vs strong noise

s=100
nobs=1000
n=6000
ind<-
c(rep(1,1000),rep(2,1000),rep(3,1000),rep(4,1000),rep(5,1000),rep(6,1000))
ind2<-
c(rep(1,1000),rep(1,1000),rep(2,1000),rep(3,1000),rep(4,1000),rep(4,1000))

yy = list()
  for(i in 1:n){
  if(ind[i]==1) yy[[i]] = c(rnorm(3*s/5,-1,0.5),rnorm(2*s/5,1,0.5))
  if(ind[i]==2) yy[[i]] = c(rnorm(2*s/5,-1,0.5),rnorm(3*s/5,1,0.5))
  if(ind[i]==3) yy[[i]] = rnorm(s,0,3)
  if(ind[i]==4) yy[[i]] = rnorm(s,0,1.5)
  if(ind[i]==5) yy[[i]] = rgumbel(s,-3,1)
  if(ind[i]==6) yy[[i]] = rgumbel(s,-3,1)
  }
dens<-list()
xabs<-seq(-7,7,length=nobs)
  for(i in 1:n){
  dens[[i]]<-density(yy[[i]],from=-7,to=7,n=nobs,bw=0.0001)$y     #bw= 0.1
, 0.0001
  }
yt=xt=vector()
  for ( i in seq(1,length(dens))){
  xx1  <- seq(1,length(dens[[i]]),len=201)
  s1<-smooth.spline(dens[[i]])
  xt<-cbind(xt,predict(s1,xx1)$y)
  yt<-cbind(yt,xx1)
  }

basis=create.fourier.basis(c(0,1),21)
pop=data2fd(xt,basisobj=basis)
coefnorms=pop$coefs/matrix(rep(pop$coefs[1,],21),nrow=21,byrow=TRUE)
coefnorms[,ind==6]=coefnorms[,ind==6]*c(rep(c(1,-1),10),1)
dis=dist_inv2(coefnorms,21)

resmds=cmdscale(dis,3,eig=T)
afm=resmds$points

par(mfrow=c(1,3))
plot(afm[,1],afm[,2],col=ind)
plot(afm[,2],afm[,3],col=ind)
plot(afm[,1],afm[,3],col=ind)

a=bagging(afm,ind2,4,30 )
mean(a)
```