Topological Data Analysis in R

Mathematics for Big Data

Jeremy Williams

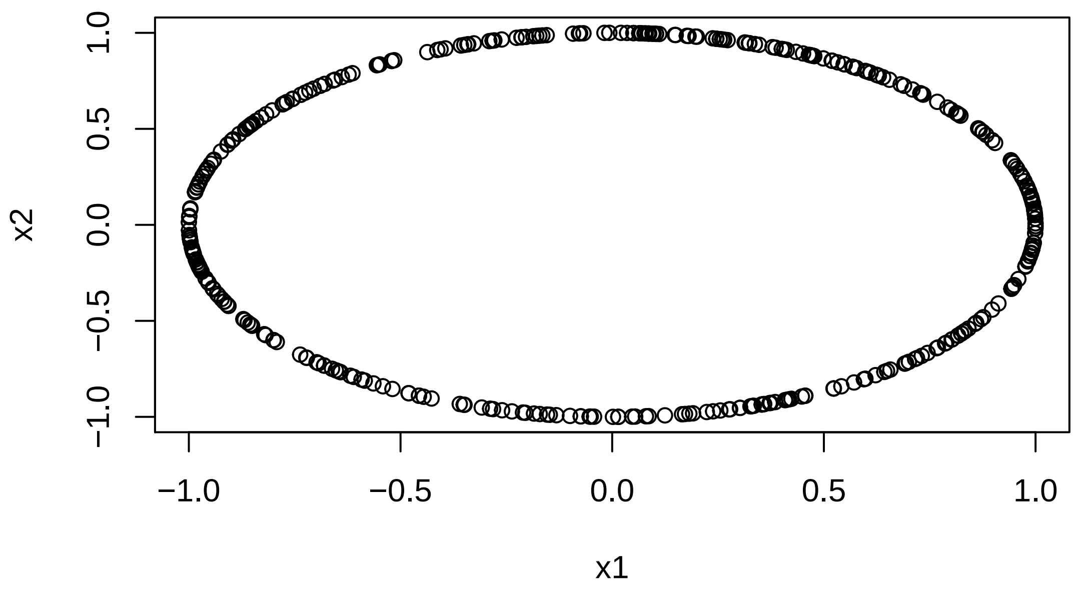
May 1, 2018

Contents

Topological Data Analysis (Practical Work)

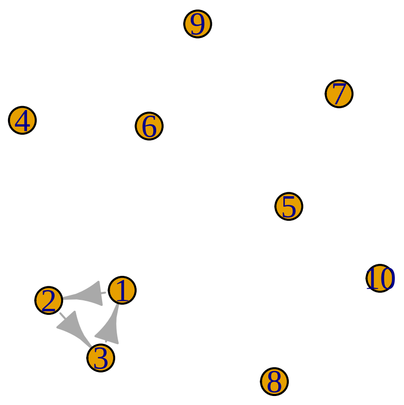
Practical 1

```
## Install code ##
#install.packages("FactoMineR", dependencies=TRUE)
#install.packages("locfit", dependencies=TRUE)
#install.packages("ks",dependencies=TRUE)
#install.packages(c('openssl', 'xml2', 'httr','git2r', 'curl', 'hunspell', lintr', 'rversions', 'covr',
#install.packages("rgl",dependencies=TRUE)
#install.packages("doParallel",dependencies=TRUE)
#install.packages("TDA",dependencies=TRUE)
#install.packages("TDAmapper",dependencies=TRUE)
#install.packages("networkD3",dependencies=TRUE)
## End of Install code ##

## Library code
suppressMessages(suppressWarnings(library("FNN")))
suppressMessages(suppressWarnings(library("TDA")))
## End Library code

## R code ##
circleSample <- circleUnif(n = 400, r = 1)
plot(circleSample)
```

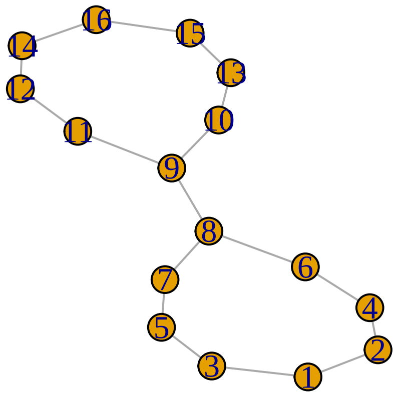


```
suppressMessages(suppressWarnings(library(rgl)))
torusSample <- torusUnif(n = 10000, a = 1.8, c = 5)
plot3d(torusSample)
```

```r
suppressMessages(suppressWarnings(library(igraph)))
g1 <- graph( edges=c(1,2, 2,3, 3, 1), n=10 )
plot(g1)
```



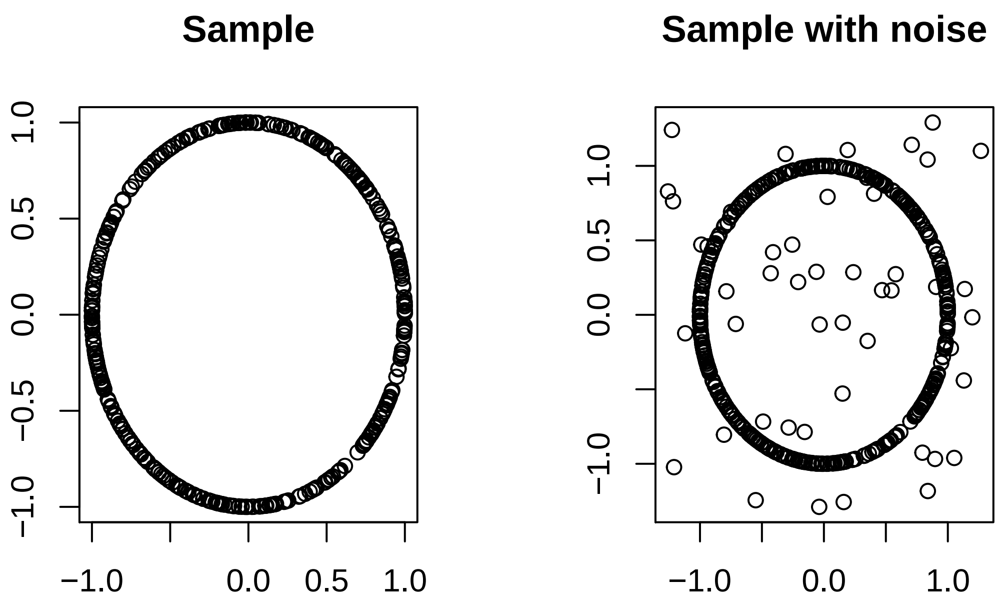
```r
suppressMessages(suppressWarnings(library(TDAmapper)))
m1 <- mapper1D(
  distance_matrix = dist(data.frame( x=2*cos(0.5*(1:100)),
  y=sin(1:100) )),
  filter_values = 2*cos(0.5*(1:100)),
  num_intervals = 10,
  percent_overlap = 50,
  num_bins_when_clustering = 10)

suppressMessages(suppressWarnings(library(igraph)))
g1 <- graph.adjacency(m1$adjacency, mode="undirected")
plot(g1, layout = layout.auto(g1) )
```



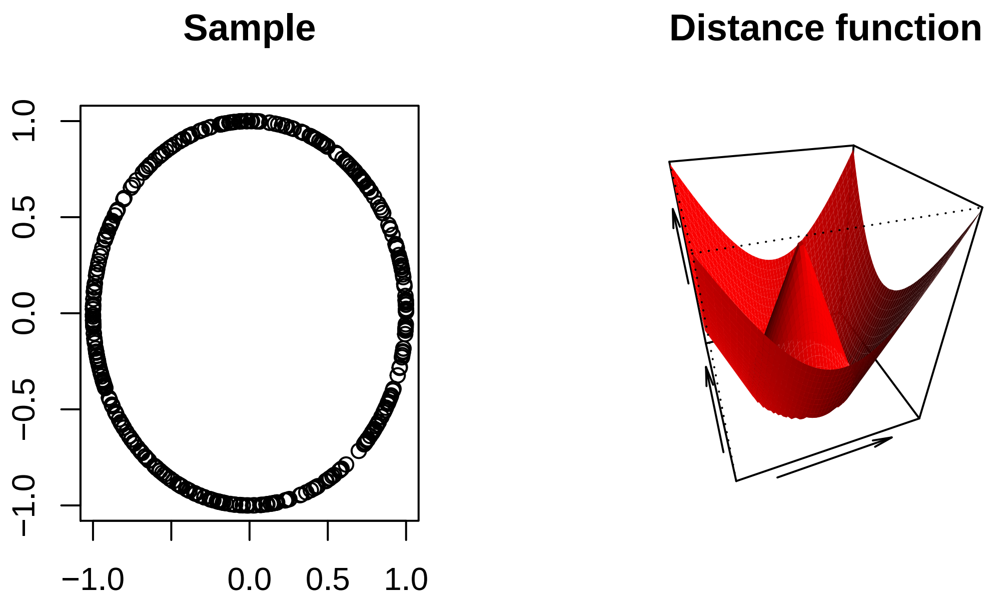
```
## End of R code ##
```

Practical 2

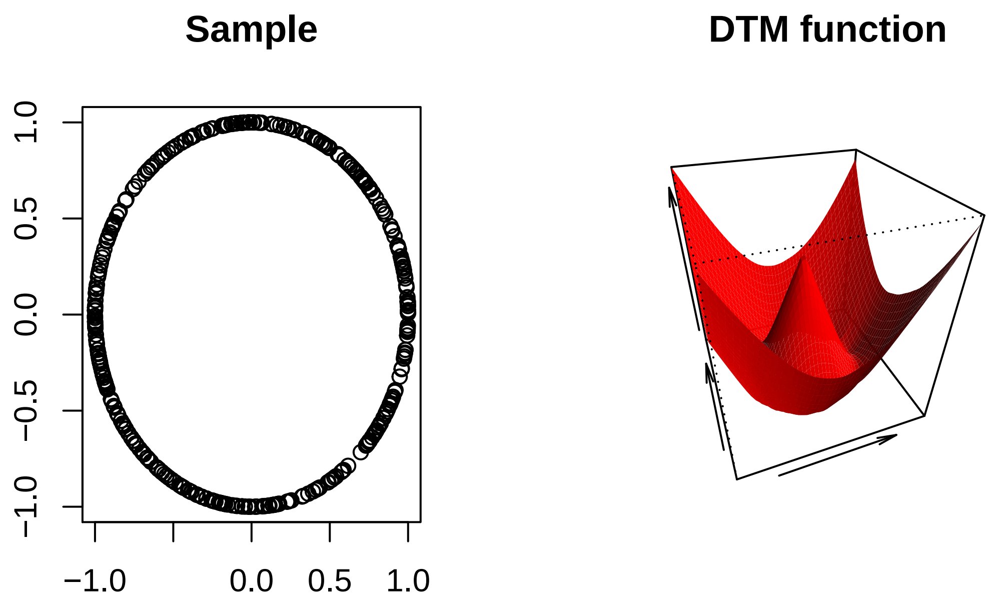
```
## Install code ##
#install.packages("FactoMineR", dependencies=TRUE)
#install.packages("locfit", dependencies=TRUE)
#install.packages("ks",dependencies=TRUE)
#install.packages(c('openssl', 'xml2', 'httr','git2r', 'curl', 'hunspell', lintr', 'rversions', 'covr',
#install.packages("rgl",dependencies=TRUE)
#install.packages("doParallel",dependencies=TRUE)
#install.packages("TDA",dependencies=TRUE)
#install.packages("TDAmapper",dependencies=TRUE)
#install.packages("networkD3",dependencies=TRUE)
## End of Install code ##

## Library code
suppressMessages(suppressWarnings(library("FNN")))
suppressMessages(suppressWarnings(library("TDA")))
## End Library code

X <- circleUnif(n=400,r=1)
xrand <- runif(50,min=-1.3,max=1.3); yrand <- runif(50,min=-1.3,max=1.3)
Y = data.frame(x1=xrand,x2=yrand)
Xnoise <- rbind(X,Y)

# Plot both data
par(mfrow=c(1,2))
plot(X,xlab="",ylab="",main="Sample")
plot(Xnoise,xlab="",ylab="",main="Sample with noise")
```



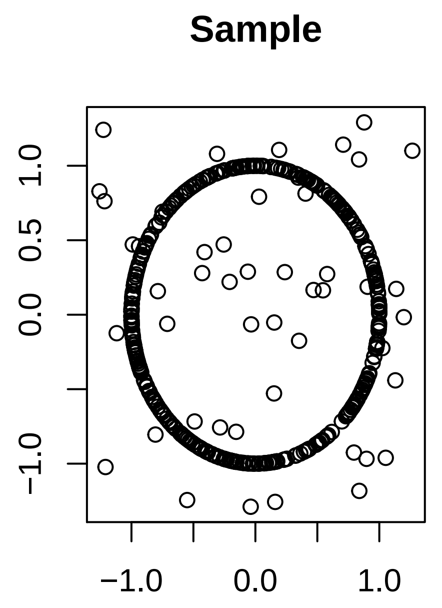
```r
# Define some parameters
Xlim <- c(-1.6,1.6)
Ylim <- c(-1.6,1.6)
by <- 0.065
Xseq <- seq(from = Xlim[1], to= Xlim[2], by = by)
Yseq <- seq(from = Ylim[1], to= Ylim[2], by = by)
Grid <- expand.grid(Xseq,Yseq)

# Use as filter function the distance function on X
distance <- distFct(X = X, Grid = Grid)
par(mfrow = c(1,2))
plot(X,xlab="",ylab="",main="Sample")
persp(x=Xseq,y=Yseq,z=matrix(distance,nrow=length(Xseq),ncol=length(Yseq)),
      xlab="",ylab="",zlab="", main="Distance function",
      theta=-20,phi=35,scale=FALSE,
      expand=3, col="red",border=NA, ltheta=50,shade=0.5)
```

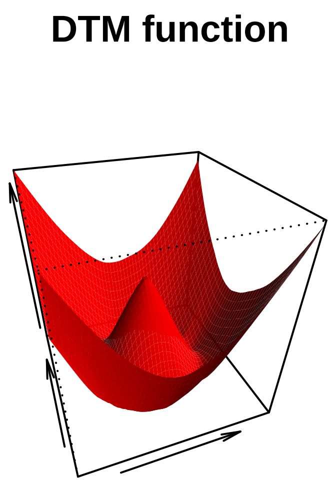


```r
# Use as filter function the DTM function on X
k0 <- 40 # 40 nearest points
m0 <- k0/length(X[,1])
DTM <- dtm(X = X, Grid = Grid, m0 = m0)
par(mfrow = c(1,2))
plot(X,xlab="",ylab="",main="Sample")
persp(x=Xseq,y=Yseq,z=matrix(DTM,nrow=length(Xseq),ncol=length(Yseq)),
      xlab="",ylab="",zlab="", main="DTM function",
      theta=-20,phi=35,scale=FALSE,
      expand=3, col="red",border=NA, ltheta=50,shade=0.5)
```

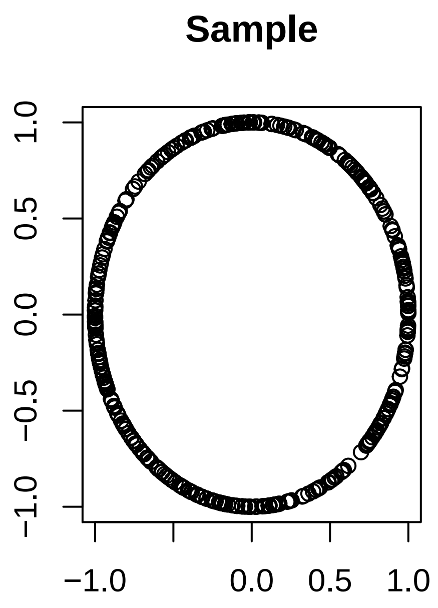


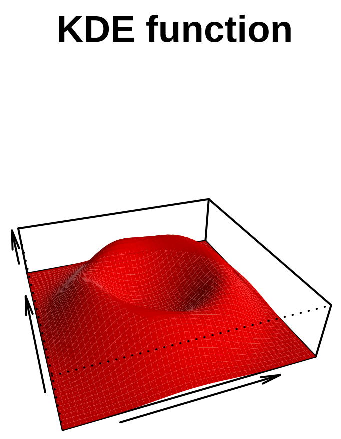
```
# Use as filter function the DTM function on Xnoise
k0 <- 40 # 40 nearest points
m0 <- k0/length(X[,1])
DTM <- dtm(X = Xnoise, Grid = Grid, m0 = m0)
par(mfrow = c(1,2))
plot(Xnoise,xlab="",ylab="",main="Sample")
persp(x=Xseq,y=Yseq,z=matrix(DTM,nrow=length(Xseq),ncol=length(Yseq)),
      xlab="",ylab="",zlab="", main="DTM function",
      theta=-20,phi=35,scale=FALSE,
      expand=3, col="red",border=NA, ltheta=50,shade=0.5)
```



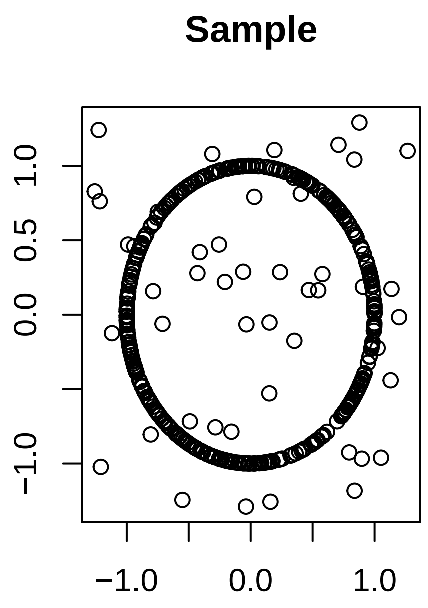


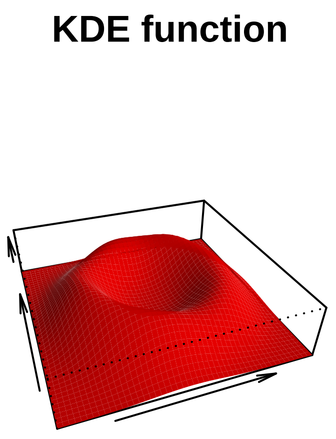
```
# Use as filter function the KDE function on X
h <- 0.3
KDE <- kde(X = X, Grid = Grid, h = h)
par(mfrow = c(1,2))
plot(X,xlab="",ylab="",main="Sample")
persp(x=Xseq,y=Yseq,z=matrix(KDE,nrow=length(Xseq),ncol=length(Yseq)),
      xlab="",ylab="",zlab="", main="KDE function",
      theta=-20,phi=35,scale=FALSE,
      expand=3, col="red",border=NA, ltheta=50,shade=0.5)
```



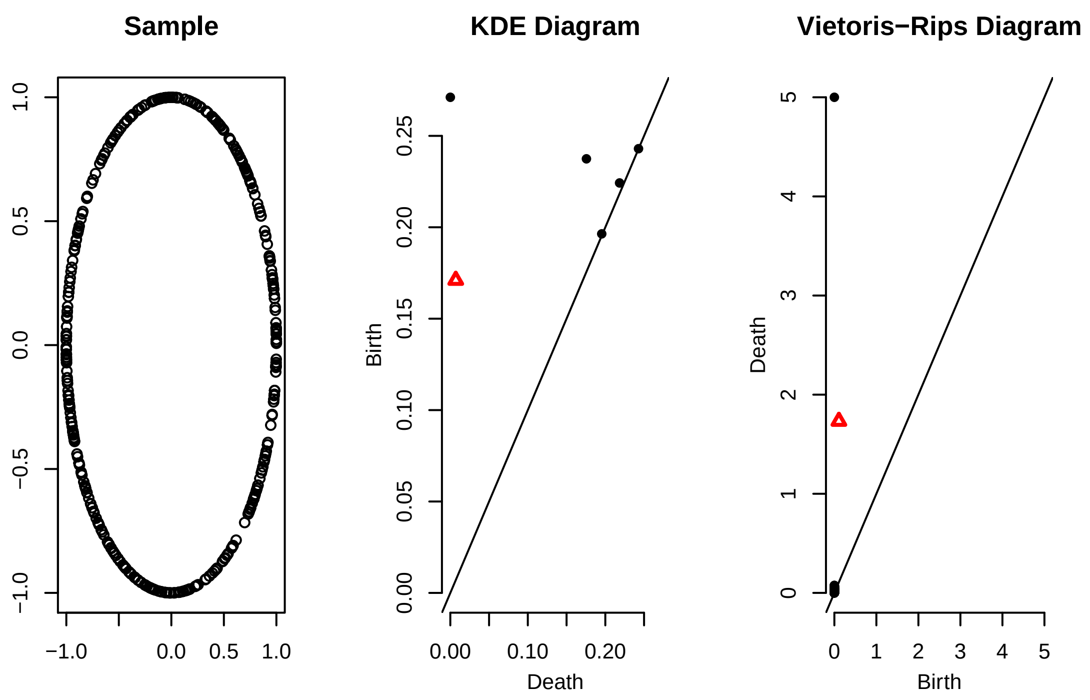


```
# Use as filter function the KDE function on Xnoise
h <- 0.3 #
KDE <- kde(X = Xnoise, Grid = Grid, h = h)
par(mfrow = c(1,2))
plot(Xnoise,xlab="",ylab="",main="Sample")
persp(x=Xseq,y=Yseq,z=matrix(KDE,nrow=length(Xseq),ncol=length(Yseq)),
      xlab="",ylab="",zlab="", main="KDE function",
      theta=-20,phi=35,scale=FALSE,
      expand=3, col="red",border=NA, ltheta=50,shade=0.5)
```

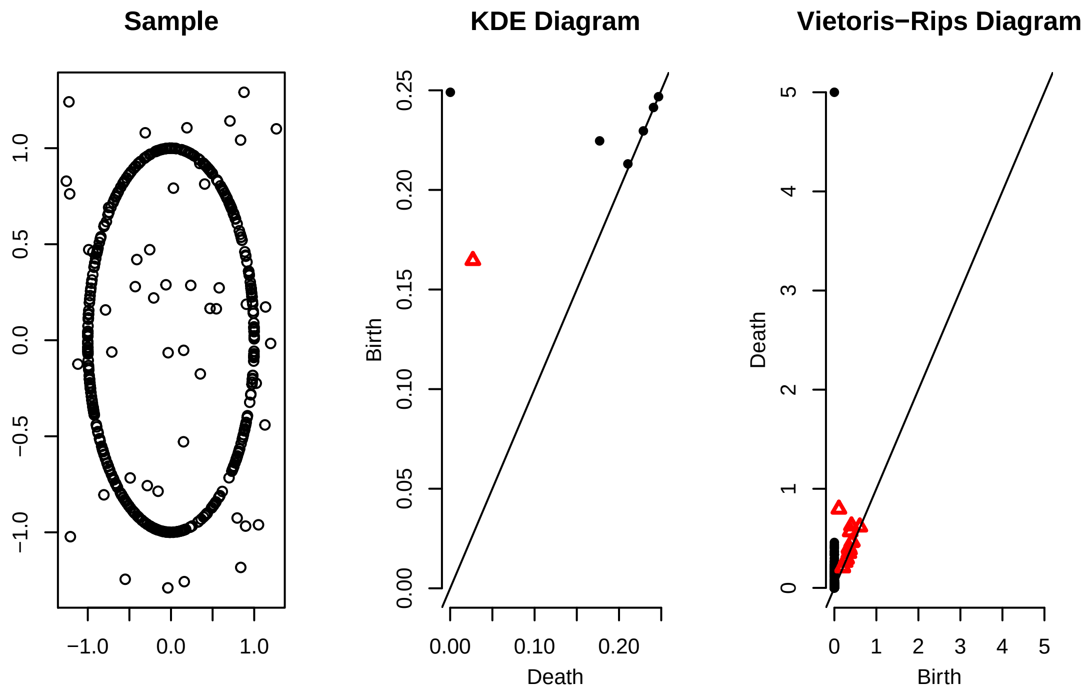




```
# Persitent homology on X with KDE function and VR
par(mfrow = c(1,3))
plot(X,xlab="",ylab="",main="Sample")
DiagKDE <- gridDiag(X=X,
                    FUN = kde, h=0.3, sublevel=FALSE,
                    lim= cbind(Xlim,Ylim), by=by,
                    library="Dionysus",printProgress = FALSE)
plot(x = DiagKDE[["diagram"]],main="KDE Diagram")
DiagVR <- ripsDiag(X=X,
                   maxdimension = 1, maxscale = 5, dist = 'euclidean',
                   library="GUDHI",printProgress = FALSE)
plot(x = DiagVR[["diagram"]],main="Vietoris-Rips Diagram")
```



```
# Persitent homology on Xnoise with KDE function and VR
par(mfrow = c(1,3))
plot(Xnoise,xlab="",ylab="",main="Sample")
DiagKDE <- gridDiag(X=Xnoise,
                    FUN = kde, h=0.3, sublevel=FALSE,
                    lim= cbind(Xlim,Ylim), by=by,
                    library="Dionysus",printProgress = FALSE)
plot(x = DiagKDE[["diagram"]],main="KDE Diagram")
DiagVR <- ripsDiag(X=Xnoise,
                   maxdimension = 1, maxscale = 5, dist = 'euclidean',
                   library="GUDHI",printProgress = FALSE)
plot(x = DiagVR[["diagram"]],main="Vietoris-Rips Diagram")
```


Sample
KDE Diagram
Vietoris−Rips Diagram
−1.0
0.0
1.0
−1.0
−0.5
0.0
0.5
1.0
Birth
Death
0.00
0.05
0.10
0.15
0.20
0.25
0.00
0.10
0.20
Death
Birth
0
1
2
3
4
5