R Package TDA for Topological Data Analysis

Jisu KIM

Carnege Mellon University

June 13, 2018

Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology

Statistical Inference on Persistence Homology

For Windows and Mac, TDA can be easily installed.

```r
if (!require(package = "TDA")) {
  install.packages(pkgs = "TDA")
}
```

For Linux, you need to install several libraries first, and then install TDA.

- You need to install libraries gmp and mpfr.
- Then you need to install required R package FNN, igraph, and scales.
- Then you can install R package TDA.

```r
if (!require(package = "FNN")) {
  install.packages(pkgs = "FNN")
}
if (!require(package = "igraph")) {
  install.packages(pkgs = "igraph")
}
if (!require(package = "scales")) {
  install.packages(pkgs = "scales")
}
if (!require(package = "TDA")) {
  install.packages(pkgs = "TDA")
}
```

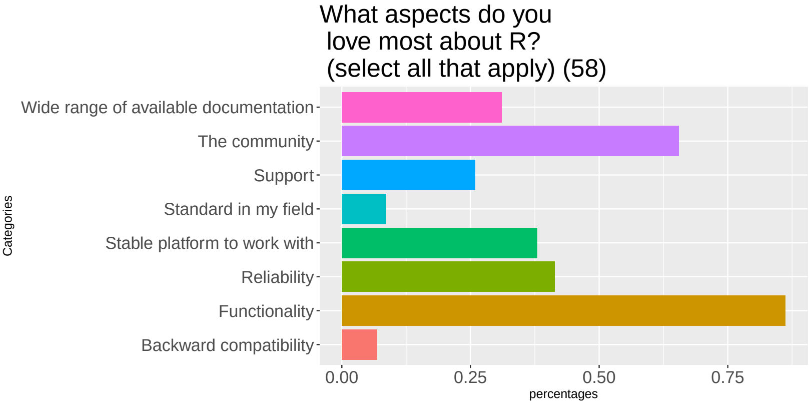
Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Sample on manifolds, Distance Functions, and Density Estimators

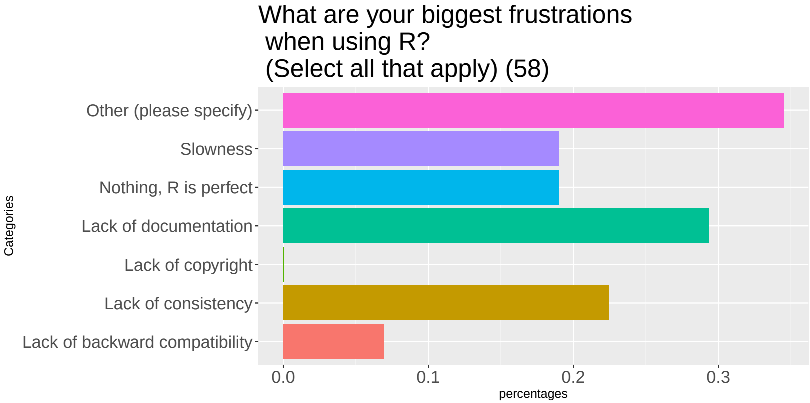
Persistent Homology

Statistical Inference on Persistence Homology



[1]

[1] Willems [2013]



[2]

[2] Willems [2013]

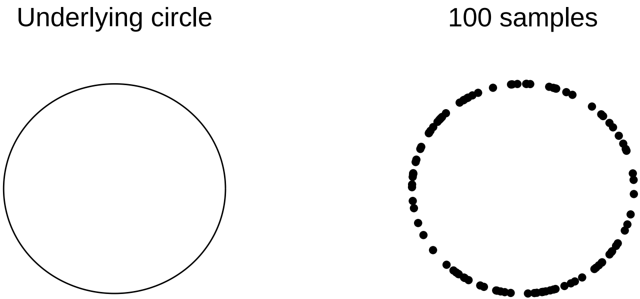
R is ideal for educational purpose.

- R is a programming language for statistical computing and graphics.
- Many packages for statistical computing.
- Easy to make (interactive) plots.
- Easy to install and use.
- Platform independent.
-
- ... but slow.

R Package TDA provides an R interface for C++ libraries for Topological Data Analysis.

- website: `https://cran.r-project.org/web/packages/TDA/index.html`
- Author: Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, Clément Maria, and Vincent Rouvreau.
- R has short development time, while C/C++ has short execution time.
- R package TDA provides an R interface for C++ library GUDHI/Dionysus/PHAT, which are for Topological Data Analysis.

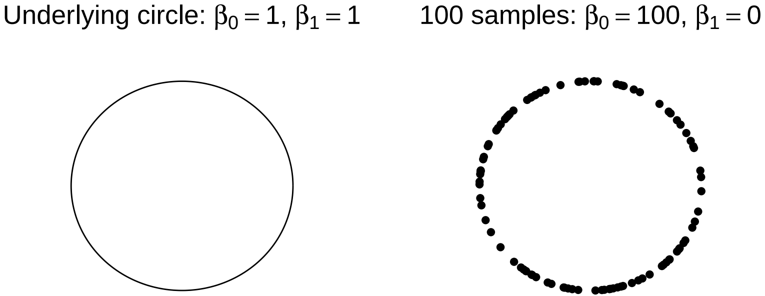
When analyzing data, we prefer robust features where features of the underlying manifold can be inferred from features of finite samples.



Underlying circle

100 samples

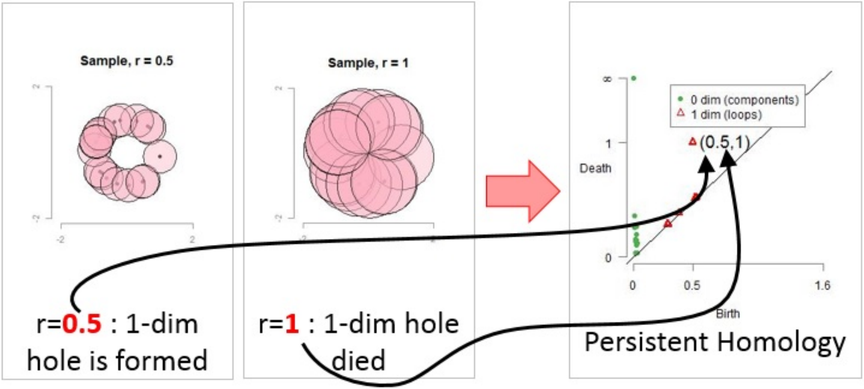
Homology of finite sample is different from homology of underlying manifold, hence it cannot be directly used for the inference.



Underlying circle: $\beta_0 = 1$, $\beta_1 = 1$

100 samples: $\beta_0 = 100$, $\beta_1 = 0$

Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.

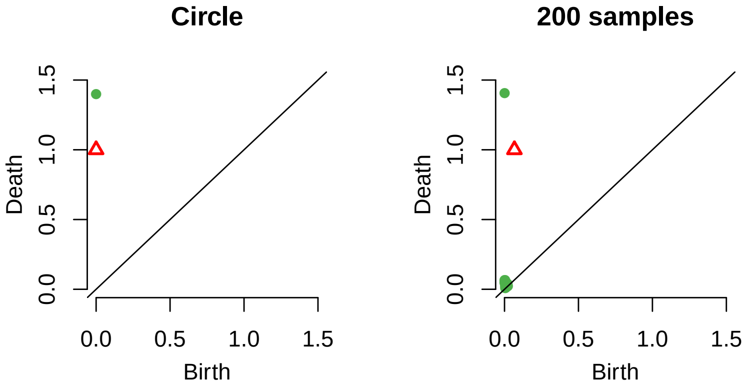


r=**0.5** : 1-dim hole is formed

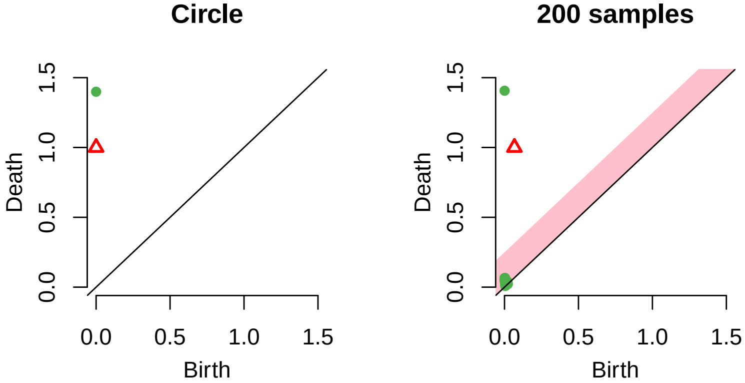
r=**1** : 1-dim hole died

Persistent Homology

Persistent homology of the underlying manifold can be inferred from persistent homology of finite samples.



Confidence band for persistent homology separates homological signal from homological noise.



Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Sample on manifolds, Distance Functions, and Density Estimators

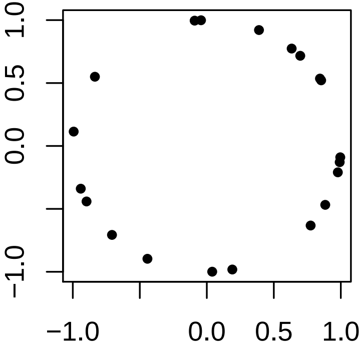
Persistent Homology

Statistical Inference on Persistence Homology

R Package TDA provides a function to sample on a circle.

The function circleUnif() generates n sample from the uniform distribution on the circle in $\mathbb{R}^2$ with radius r.

```r
circleSample <- circleUnif(n = 20, r = 1)
plot(circleSample, xlab = "", ylab = "", pch = 20)
```



R Package TDA provides density functions over a grid.

Suppose $n = 400$ points are generated from the unit circle, and grid of points are generated.

```r
X <- circleUnif(n = 400, r = 1)

lim <- c(-1.7, 1.7)
by <- 0.05
margin <- seq(from = lim[1], to = lim[2], by = by)
Grid <- expand.grid(margin, margin)
```

R Package TDA provides density functions over a grid.

The Gaussian Kernel Density Estimator (KDE) $\hat{p}_h : \mathbb{R}^d \to [0, \infty)$ is defined as

$$\hat{p}_h(y) = \frac{1}{n(\sqrt{2\pi}h)^d} \sum_{i=1}^{n} \exp\left(\frac{-\|y - x_i\|_2^2}{2h^2} \right),$$

where h is a smoothing parameter.
The function kde() computes the KDE function $\hat{p}_h$ on a grid of points.

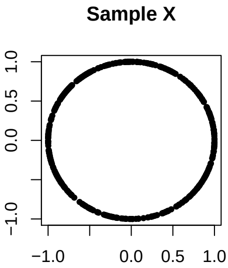
```
h <- 0.3
KDE <- kde(X = X, Grid = Grid, h = h)

par(mfrow = c(1,2))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
persp(x = margin, y = margin,
  z = matrix(KDE, nrow = length(margin), ncol = length(margin)),
  xlab = "", ylab = "", zlab = "", theta = -20, phi = 35, scale = FALSE,
  expand = 3, col = "red", border = NA, ltheta = 50, shade = 0.5,
  main = "KDE")
```

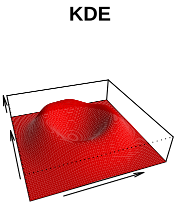
R Package TDA provides density functions over a grid.

The Gaussian Kernel Density Estimator (KDE) $\hat{p}_h : \mathbb{R}^d \to [0, \infty)$ is defined as

$$\hat{p}_h(y) = \frac{1}{n(\sqrt{2\pi}h)^d} \sum_{i=1}^{n} \exp\left(\frac{-\|y - x_i\|_2^2}{2h^2} \right),$$

where h is a smoothing parameter.
The function kde() computes the KDE function $\hat{p}_h$ on a grid of points.





Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology

Statistical Inference on Persistence Homology

R Package TDA computes Persistent Homology over a grid.

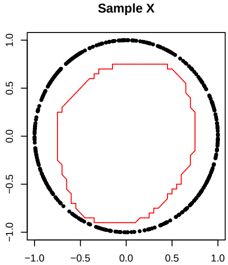
- The function gridDiag() computes the persistence diagram of sublevel (and superlevel) sets of the input function.
  - gridDiag() evaluates the real valued input function over a grid.
  - gridDiag() constructs a filtration of simplices using the values of the input function.
  - gridDiag() computes the persistent homology of the filtration.
- The user can choose to compute persistent homology using either C++ library GUDHI, Dionysus, or PHAT.

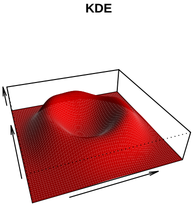
R Package TDA computes Persistent Homology over a grid.

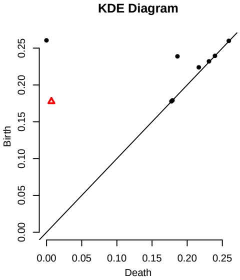
```r
DiagGrid <- gridDiag(X = X, FUN = kde, lim = c(lim, lim), by = by,
    sublevel = FALSE, library = "Dionysus", location = TRUE,
    printProgress = FALSE, h = h)

par(mfrow = c(1,3))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
one <- which(DiagGrid[["diagram"]][, 1] == 1)
for (i in seq(along = one)) {
  for (j in seq_len(dim(DiagGrid[["cycleLocation"]][[one[i]]])[1])) {
    lines(DiagGrid[["cycleLocation"]][[one[i]]][j, , ], pch = 19, cex = 1,
        col = i + 1)
  }
}
persp(x = margin, y = margin,
  z = matrix(KDE, nrow = length(margin), ncol = length(margin)),
  xlab = "", ylab = "", zlab = "", theta = -20, phi = 35, scale = FALSE,
  expand = 3, col = "red", border = NA, ltheta = 50, shade = 0.9,
  main = "KDE")
plot(x = DiagGrid[["diagram"]], main = "KDE Diagram")
```

R Package TDA computes Persistent Homology over a grid.

- The function gridDiag() computes the persistent homology of sublevel (and superlevel) sets of the input function.
    - gridDiag() evaluates the real valued input function over a grid.
    - gridDiag() constructs a filtration of simplices using the values of the input function.
    - gridDiag() computes the persistent homology of the filtration.
- The user can choose to compute persistent homology using either Dionysus library or PHAT library.







R Package TDA computes Rips Persistent Homology.

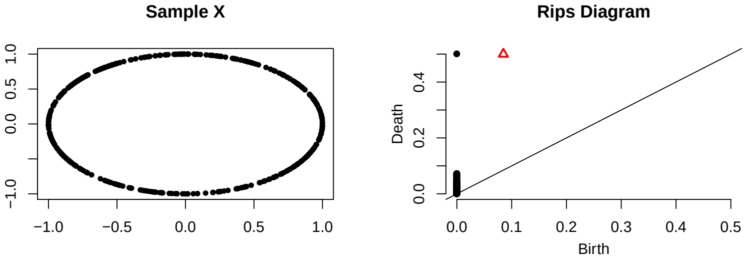
- Rips complex consists of simplices whose pairwise distances of vertices are at most ϵ apart, i.e.

$$R(X,\epsilon) = \left\{ [X_{n_1}, \ldots, X_{n_r}] :\ d(X_{n_i}, X_{n_j}) \leq \epsilon \right\}.$$

- Rips filtration is formed by Rips complices with gradually increasing ϵ.
- The function ripsDiag() computes the persistence diagram of the Rips filtration built on top of a point cloud.
  - ripsDiag() constructs the Rips filtration using the data points.
  - ripsDiag() computes the persistent homology of the Rips filtration.
- The user can choose to compute persistent homology using either C++ library GUDHI, Dionysus, or PHAT.

R Package TDA computes Rips Persistent Homology.

```
DiagRips <- ripsDiag(X = X, maxdimension = 1, maxscale = 0.5,
    library = c("GUDHI", "Dionysus"), location = TRUE)

par(mfrow = c(1,2))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
plot(x = DiagRips[["diagram"]], main = "Rips Diagram")
```



R Package TDA builds Rips filtration.

- The function ripsFiltration() builds the Rips filtration built on top of a point cloud.
- The user can choose to compute persistent homology using either C++ library GUDHI, Dionysus, or PHAT.

```r
FltRips <- ripsFiltration(X = X, maxdimension = 1, maxscale = 0.5,
    library = "GUDHI")
```

R Package TDA builds filtration from function values.

- The function funFiltration() builds the filtration from the complex and the function values.

```
h <- 0.3
KDEx <- kde(X = X, Grid = X, h = h)

FltFun <- funFiltration(FUNvalues = KDEx, cmplx = FltRips[["cmplx"]],
    sublevel = FALSE)
```

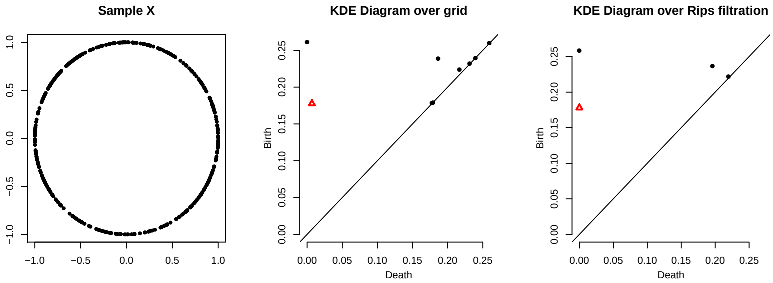
R Package TDA computes Persistent Homology from filtration.

- The function filtrationDiag() computes the persistent homology from the filtration.

```r
DiagFltFun <- filtrationDiag(filtration = FltFun, maxdimension = 1,
    library = "Dionysus", location = TRUE, printProgress = FALSE)

par(mfrow = c(1,3))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
plot(x = DiagGrid[["diagram"]], main = "KDE Diagram over grid")
plot(x = DiagFltFun[["diagram"]], diagLim = c(0, 0.27),
    main = "KDE Diagram over Rips filtration")
```

R Package TDA computes Persistent Homology from filtration.

- The function filtrationDiag() computes the persistent homology from the filtration.



Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology

Statistical Inference on Persistence Homology

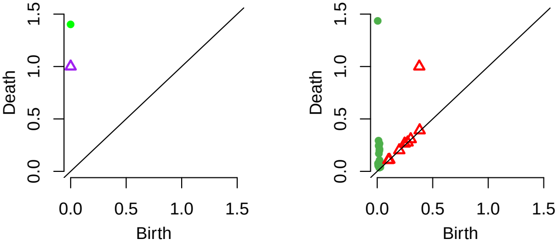
Bottleneck distance gives a metric on the space of persistent homology.

Definition

Let $D_1,\ D_2$ be multiset of points. Bottleneck distance is defined as

$$W_\infty(D_1,\ D_2) = \inf_{\gamma} \sup_{x \in \mathbf{D_1}} \|x - \gamma(x)\|_\infty,$$

where γ ranges over all bijections from D_1 to D_2.



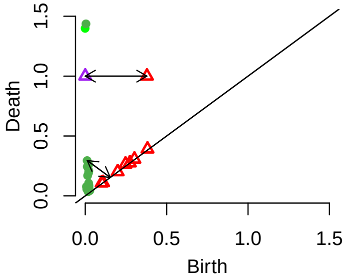
Bottleneck distance gives a metric on the space of persistent homology.

Definition

Let D_1, D_2 be multiset of points. Bottleneck distance is defined as

$$W_\infty(D_1,\, D_2) = \inf_{\gamma} \sup_{x \in D_{\mathbf{1}}} \|x - \gamma(x)\|_\infty,$$

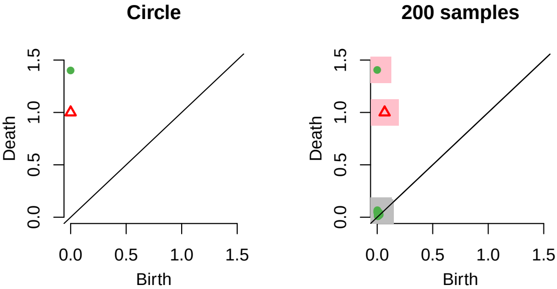
where γ ranges over all bijections from D_1 to D_2.



Confidence band for the persistent homology is a random quantity containing the persistent homology with high probability.

Let M be a compact manifold, and $X = \{X_1, \cdots, X_n\}$ be n samples. Let f_M and f_X be corresponding functions whose persistent homology is of interest. Given the significance level $\alpha \in (0, 1)$, $(1 - \alpha)$ confidence band $c_n = c_n(X)$ is a random variable satisfying

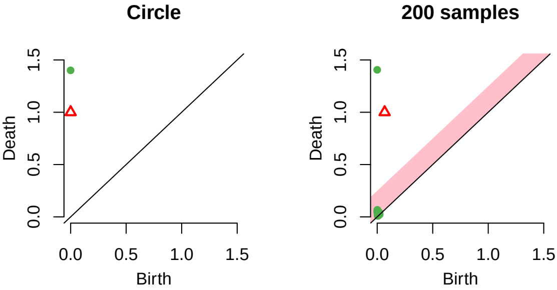
$$\mathbb{P}\left(W_\infty(Dgm(f_M),\, Dgm(f_X)) \leq c_n\right) \geq 1 - \alpha.$$



Confidence band for the persistent homology is a random quantity containing the persistent homology with high probability.

Let M be a compact manifold, and $X = \{X_1, \cdots, X_n\}$ be n samples. Let f_M and f_X be corresponding functions whose persistent homology is of interest. Given the significance level $\alpha \in (0, 1)$, $(1-\alpha)$ confidence band $c_n = c_n(X)$ is a random variable satisfying

$$\mathbb{P}\left(W_\infty(Dgm(f_M),\, Dgm(f_X)) \leq c_n\right) \geq 1 - \alpha.$$



Confidence band for the persistent homology can be computed using the bootstrap algorithm.

1. Given a sample $X = \{x_1, \ldots, x_n\}$, compute the kernel density estimator $\hat{p}_h$.
2. Draw $X^* = \{x_1^*, \ldots, x_n^*\}$ from $X = \{x_1, \ldots, x_n\}$ (with replacement), and compute $\theta^* = \sqrt{n}||\hat{p}_h^*(x) - \hat{p}_h(x)||_\infty$, where $\hat{p}_h^*$ is the density estimator computed using X^*.
3. Repeat the previous step B times to obtain $\theta_1^*, \ldots, \theta_B^*$
4. Compute $q_\alpha = \inf\left\{q : \frac{1}{B}\sum_{j=1}^{B} I(\theta_j^* \geq q) \leq \alpha\right\}$
5. The $(1-\alpha)$ confidence band for $\mathbb{E}[\hat{p}_h]$ is $\left[\hat{p}_h - \frac{q_\alpha}{\sqrt{n}}\,,\, \hat{p}_h + \frac{q_\alpha}{\sqrt{n}}\right]$.

R Package TDA computes the bootstrap confidence band for a function.

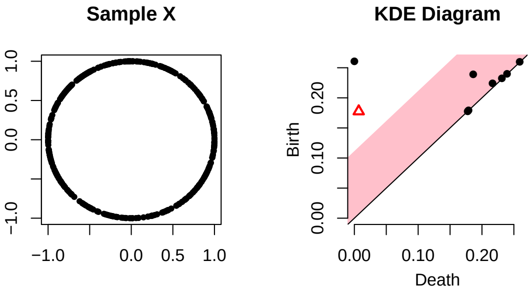
The function bootstrapBand() computes $(1-\alpha)$ bootstrap confidence band for $\mathbb{E}[\hat{p}_h]$.

```r
bandKDE <- bootstrapBand(X = X, FUN = kde, Grid = Grid, B = 20,
    parallel = FALSE, alpha = 0.1, h = h)
print(bandKDE[["width"]])

##        90%
## 0.05576625
```

The bootstrap confidence band for a function is used as the confidence band for the persistent homology.

The $(1-\alpha)$ bootstrap confidence band for $\mathbb{E}[\hat{p}_h]$ is used as the confidence band for the persistent homology.

```r
par(mfrow = c(1,2))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
plot(x = DiagGrid[["diagram"]], band = 2 * bandKDE[["width"]],
     main = "KDE Diagram")
```



Reference

Karlijn Willems. R and education: a survey on the use of r in education, 06 2013. URL https://www.datacamp.com/community/blog/survey-on-r-and-education.

Thank you!