README

A "How To"

Fred Kaesmann, Dani Rosenberg and Kun Woo Lee

April 4, 2020

How to Handle our Package

Welcome to the intro to

```
library(FredsVietorisRips)
```

We lovingly designed this package for use in Dr. Aaron Clark's MAT 499 Topological Data Analysis Independent Study. [1] This package was developed to replicate aspects of the Mapper Algorithm from *insert sing, carlsson et al.*

It is to be used in the following manner

Usage

Downloading

To download from Github, copy and paste the following into the console

```
if (!require(FredsVietorisRips)) install_github("ftkjr/FredsVietorisRips")
library(FredsVietorisRips)
```

Simplices, 0 and 1

To create our 1-simplices,

Step 1: We generate a data frame of random x and y coordinates

Step 2: We create a matrix of pairwise distances. Each cell i, j represents the distance from the ith point to the jth point.

Step 3: Given a distance, ϵ, we return an adjacency matrix which displays a 1 for each pair whose Euclidean Distance is within the given ϵ value

Step 4: From the adjacency matrix, we pull out the points which are adjacent. This is a list of all 1-simplexes in the dataset

Step 5: Last, we visualize the simplices using the 'ggplot2' library

[1] TDAIS for short

```r
##### Step 1: Create Data Frame ####
frame_size <- 7
df <- data.frame(
  x = runif(frame_size),
  y = runif(frame_size),
  Point = paste0("P", c(1:frame_size))
)

##### Step 2: Pairwise Distance Matrix ####

pwdmat <- Pairwisedist(df$x, df$y)
pwdmat
```

```
##           P1        P2        P3        P4         P5         P6        P7
## P1 0.0000000 0.3523618 0.7679247 0.5858046 0.83163440 0.78042370 0.8515422
## P2 0.3523618 0.0000000 0.6937876 0.3463367 0.67158113 0.60178231 0.4998649
## P3 0.7679247 0.6937876 0.0000000 0.4106228 0.19238020 0.22899319 0.8999429
## P4 0.5858046 0.3463367 0.4106228 0.0000000 0.33531864 0.26165989 0.5054511
## P5 0.8316344 0.6715811 0.1923802 0.3353186 0.00000000 0.07545697 0.7675711
## P6 0.7804237 0.6017823 0.2289932 0.2616599 0.07545697 0.00000000 0.6966016
## P7 0.8515422 0.4998649 0.8999429 0.5054511 0.76757108 0.69660164 0.0000000
```

```r
##### Step 3: Given epsilon, determine Adjacency ####
# Given epsilon
epsilon <- 0.5

# Determine Adjacency
adjacency_matrix <- AdjacencyMatrix(pwdmat, epsilon)
adjacency_matrix
```

```
##    P1 P2 P3 P4 P5 P6 P7
## P1  1  1  0  0  0  0  0
## P2  1  1  0  1  0  0  1
## P3  0  0  1  1  1  1  0
## P4  0  1  1  1  1  1  0
## P5  0  0  1  1  1  1  0
## P6  0  0  1  1  1  1  0
## P7  0  1  0  0  0  0  1
```

```r
##### Step 4: Which Points are Adjacent? ####
paired_points <- AdjacentPairs(adjacency_matrix)
paired_points
```

```
##   Point_1 Point_2 Connection group
## 1      P1      P2          1     1
## 2      P2      P4          1     2
## 3      P3      P4          1     3
## 4      P3      P5          1     4
## 5      P4      P5          1     5
## 6      P3      P6          1     6
## 7      P4      P6          1     7
## 8      P5      P6          1     8
## 9      P2      P7          1     9
```

```
##### Step 5: Plot 0 and 1 Simplices ####
Plot_1_Simplices(df$x, df$y, epsilon)
```