Assignment: Local MDS, ISOMAP and t-SNE

ZEROs in the ZIP numbers dataset

Mohana Fathollahi, Thanh Binh Viedena Vu

21 de December de 2021

0) Load data and provided functions

ZEROs in the ZIP numbers dataset

Consider the ZIP number data set, from the book of Hastie et al. (2009). Read the training data set (in the file zip.train) and select only the zeros. There are n = 1194 digits corresponding to ZEROs in the data set.

```
zip.train <- read.table("zip.train")
zip.train.0 <-zip.train[zip.train[,'V1']==0,]
```

Function to plot a digit

```
plot.zip <- function(x,use.first=FALSE){
  x<-as.numeric(x)
  if (use.first){
    x.mat <- matrix(x,16,16)
  } else {
    x.mat <- matrix(x[-1],16,16)
  }
  image(1:16,1:16,x.mat[,16:1],
        col=gray(seq(1,0,l=12)))
  if (!use.first) title(x[1])
}
```

1) Local MDS for ZERO digits

You must apply Local MDS to reduce the dimensionality of this dataset using the function lmds from package stops. You have to install the library stops from this link and then to attach the library:

```
if (!require(stops, quietly=TRUE, warn.conflicts=FALSE)){
  install.packages("stops",
                   repos="http://R-Forge.R-project.org",
                   INSTALL_opts="--no-test-load")
}
```

```
##
## Attaching package: 'smacof'
```

```
## The following object is masked from 'package:base':
##
##     transform
```

```
## This build of rgl does not include OpenGL functions.  Use
##  rglwidget() to display results, e.g. via options(rgl.printRglwidget = TRUE).
```

```
##
## Attaching package: 'rgl'
```

```
## The following object is masked from 'package:plotrix':
##
##     mtext3d
```

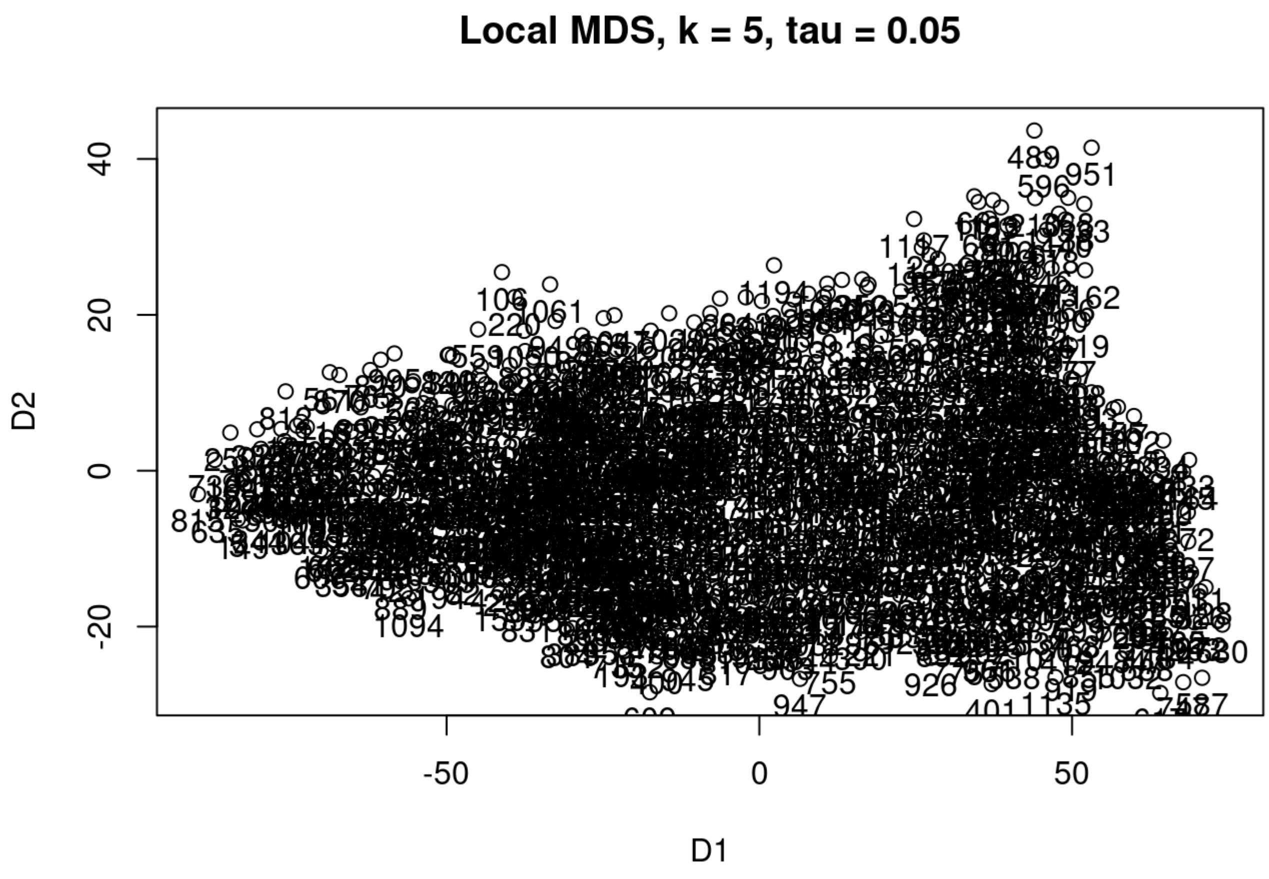
```
library(stops) # help(lmds)
```

1a) Look for 2-dimensional configuration of given data

Look for a 2-dimensional (q = 2) configuration of the data using parameters k = 5 and τ = 0.05 in lmdsfunction. Do the scatterplot of the obtained 2-dimensional configuration.

```
dist.0 <- dist(zip.train.0)
n <- dim(dist.0)[1]

k <- 5
tau <- .05
q <- 2
conf0 <- cmdscale(dist.0, k=q)$points
lmds.S.res <- lmds(as.matrix(dist.0), init=conf0, ndim=q, k=k, tau=tau, itmax = 1000)
conf.lmds.S.res <- lmds.S.res$conf
```

```
plot(conf.lmds.S.res, main=paste0("Local MDS, k = ", k, ", tau = ", tau))
text(conf.lmds.S.res[,1], conf.lmds.S.res[,2], 1:n, pos=1)
```

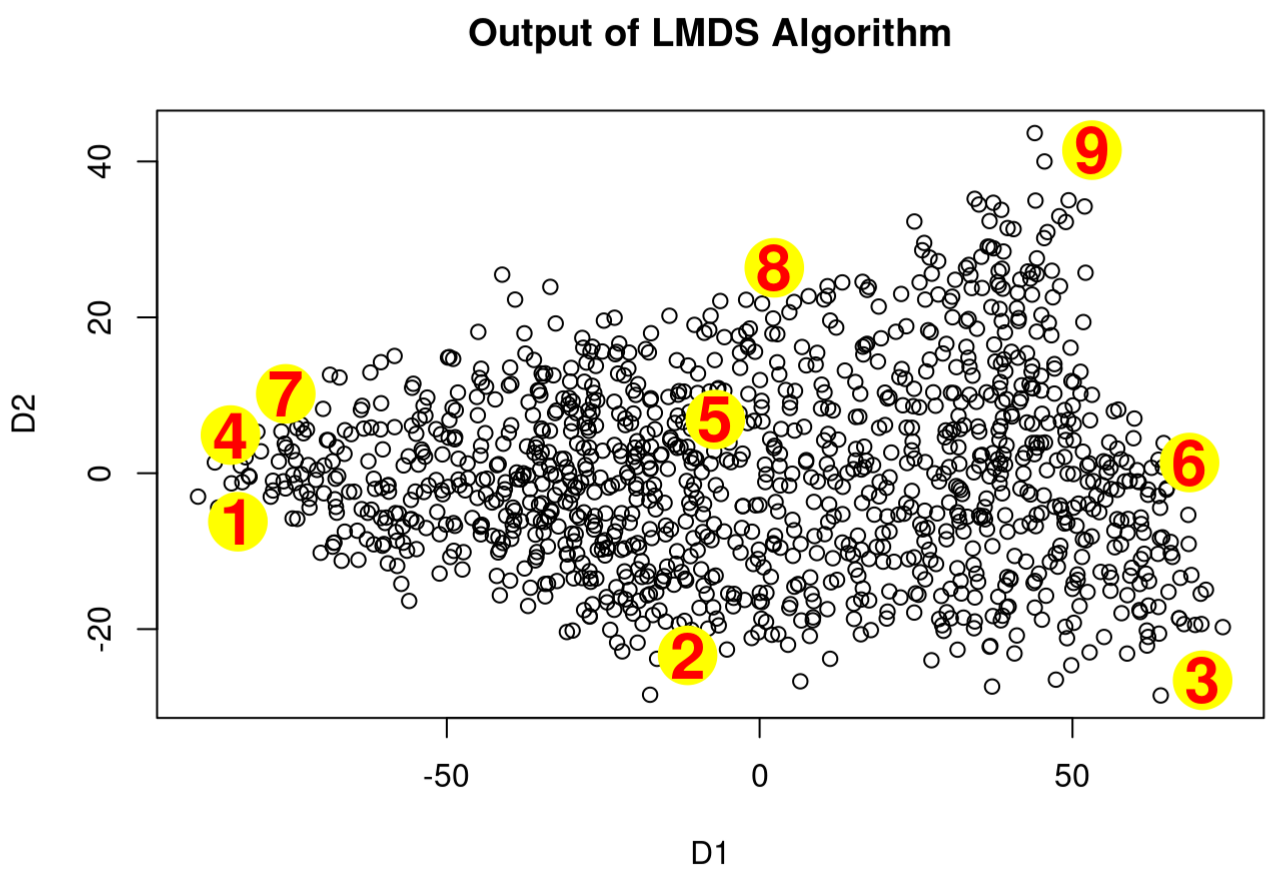


1b) Select points from scatterplot to cover variability

In the previous scatterplot, select a few points (9 points, for instance) located in such a way that they cover the variability of all the points in the scatterplot. Then use the function plot.zip to plot the ZERO digits corresponding to the selected points. The images you are plotting should allow you to give an interpretation of the 2 coordinates obtained by Local MDS (observe how the shape of ZEROs changes when moving along each directions of the scatterplot).

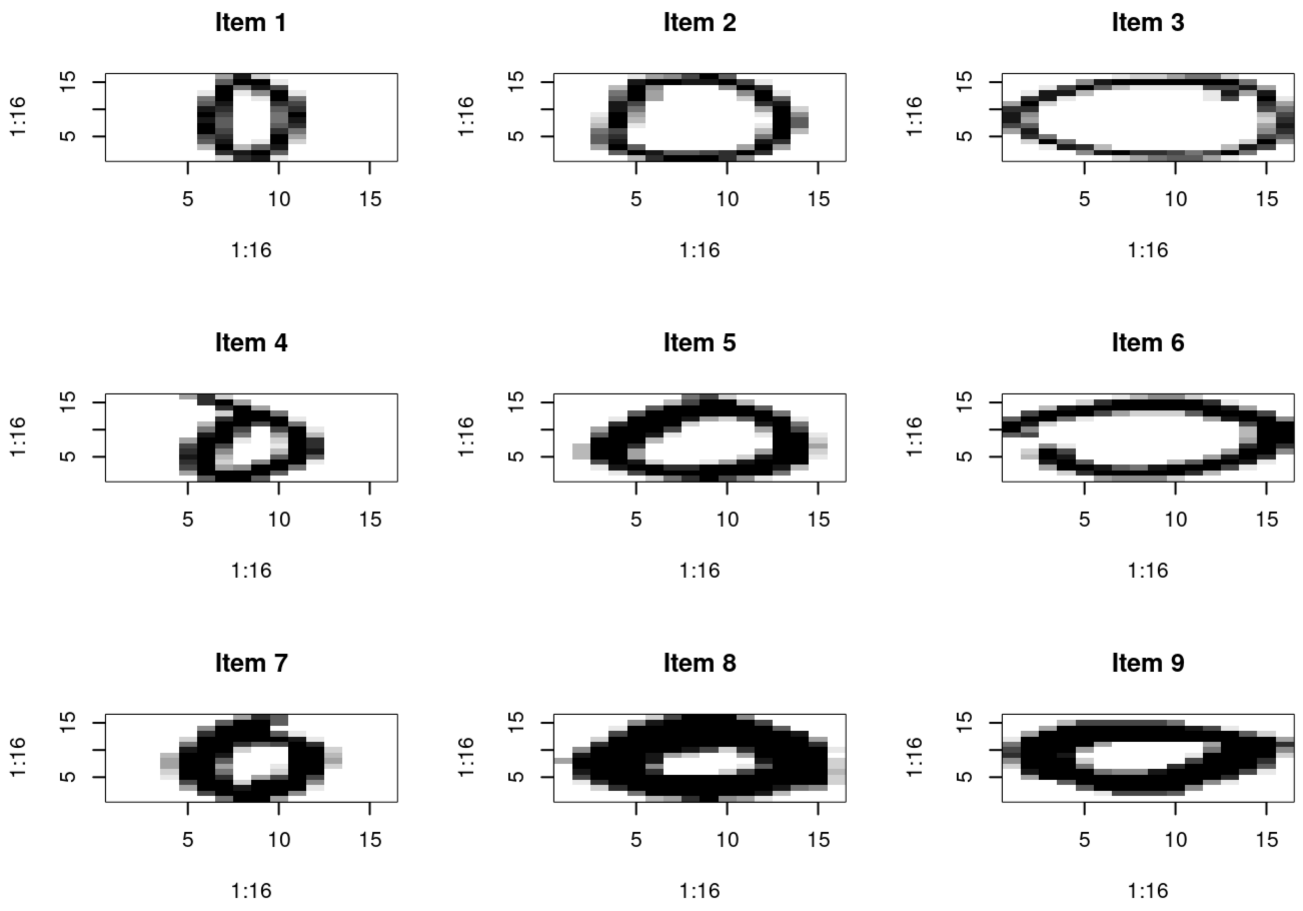
```
points.for.variability <- function(M){
  min_1 <- min(M[[1]], na.rm = TRUE)
  max_1 <- max(M[[1]], na.rm = TRUE)
  min_2 <- min(M[[2]], na.rm = TRUE)
  max_2 <- max(M[[2]], na.rm = TRUE)

  v1 <- c(min_1, (min_1 + max_1)/2, max_1)
  v2 <- c(min_2, (min_2 + max_2)/2, max_2)

  p <- expand.grid(v1,v2)
  p_id <- vector("numeric", nrow(p))
  for (id in 1:nrow(p)){
    p_id[id] <- which.min((p[id,1] - M[[1]])^2 + (p[id,2] - M[[2]])^2)
  }
  return(rbind(v1,v2,p_id))
}
```

```
M <- as.data.frame(conf.lmds.S.res)
pts.var <- points.for.variability(M)
ids <- pts.var[3,]
aux.plot <- plot(conf.lmds.S.res, main="Output of LMDS Algorithm")
points(M[ids,1],M[ids,2],col='yellow',pch=19,cex=4)
text (M[ids,1],M[ids,2],1:length(M[ids,1]), cex=2, font=2,col='red')
```



```
op <- par(mfrow=c(3,4))
draw.digits <- function(ids, M, v1, v2){
  p <- expand.grid(v1,v2)
  for (id in 1:length(M[ids,1])){
    plot.zip(zip.train.0[,-1][ids[id],],use.first = TRUE)
    title(paste("Item",id))
  }
}
par(op)
```

```
op <- par(mfrow=c(3,3))
draw.digits(ids,M,pts.var[1,],pts.var[2,])
```



```
par(op)
```

It can be observed that in LMDS, when we move in x-axies from item 1 to item 3, shape of zeros are becoming wider. Additionally, when we move in y axies, shape of zeros are becoming thicker. As a result, we expect that item 7,8 and 9 have thicker shape compare to item 1, 2 and 3.

1c) (OPTIONAL)

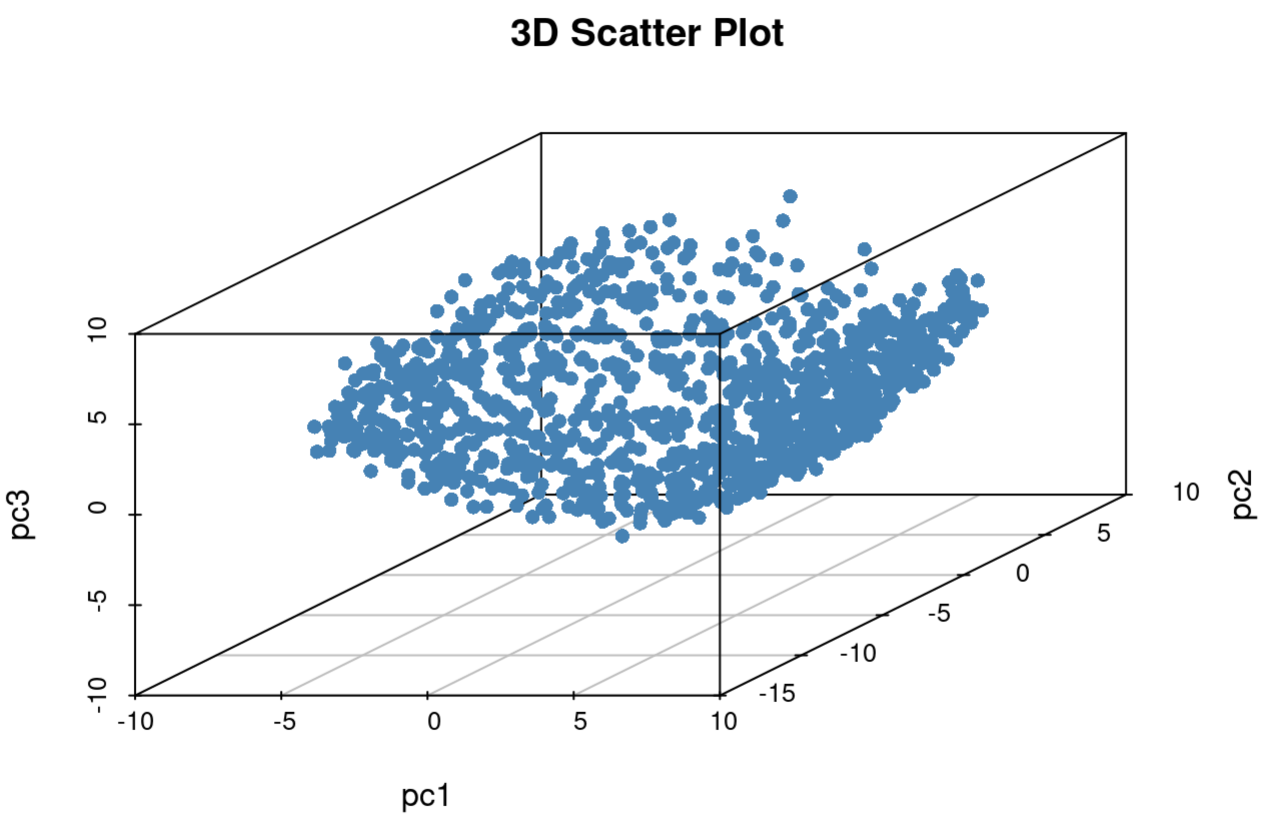
Relate the results from Local MDS with those obtained by the first 3 principal components. In particular, could you represent in any way the results obtained by Local MDS in the 3-dimensionap scatterplot pf (PC1,PC2,PC3)?

```
zip.0.PC <- princomp(zip.train.0[,-1])
```

```
install.packages("scatterplot3d") # Install
```

```
## Installing package into '/home/hamid/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library("scatterplot3d") # load
scatterplot3d(zip.0.PC$scores[,1:3],
              main="3D Scatter Plot",
              xlab = "pc1",
              ylab = "pc2",
              zlab = "pc3",pch = 16, color="steelblue")
```



2) ISOMAP for ZERO digits

a, b, c. Repeat previous points a and b (and OPTIONALLY c) but using now ISOMAP. Use function isomap from package vegan and use parameter k = 5 (instead of using ε)

2a) Look for 2-dimensional configuration of given data

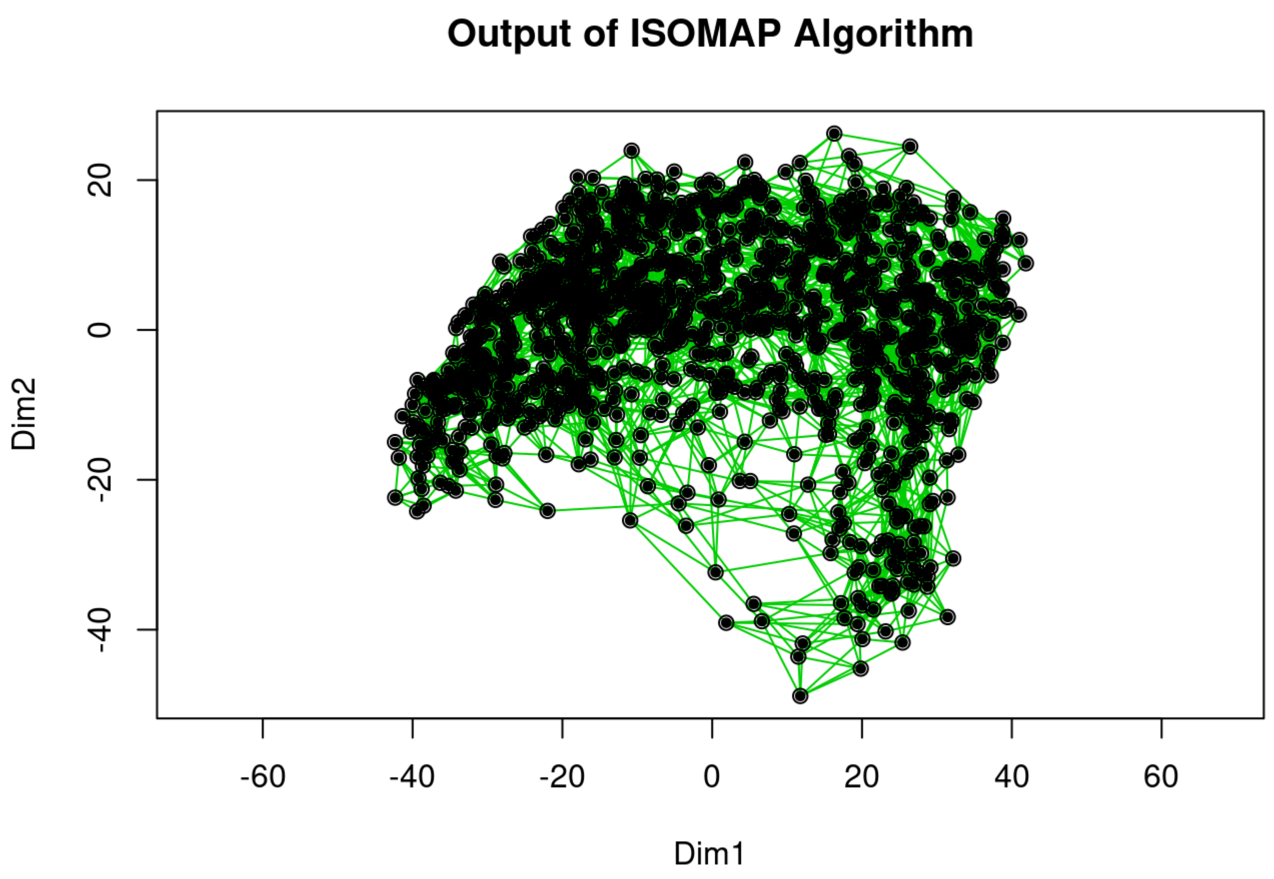
```
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

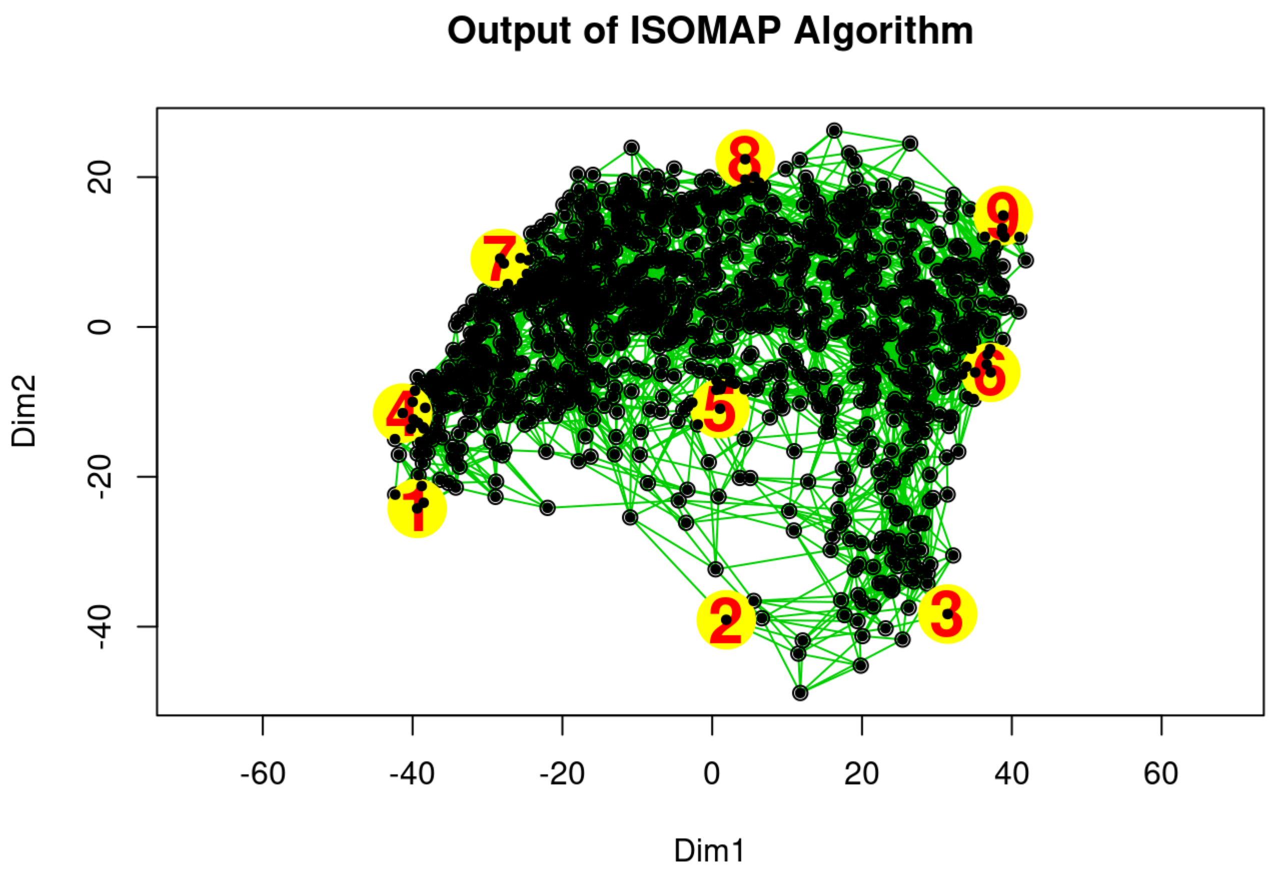
```
## This is vegan 2.5-7
```

```
D  <- dist(zip.train.0[,-1])
L <- isomap(D, ndim=2, k=5)
aux.plot <- plot(L,n.col=3,main="Output of ISOMAP Algorithm")
points(aux.plot,"sites",pch=19,cex=.6)
```

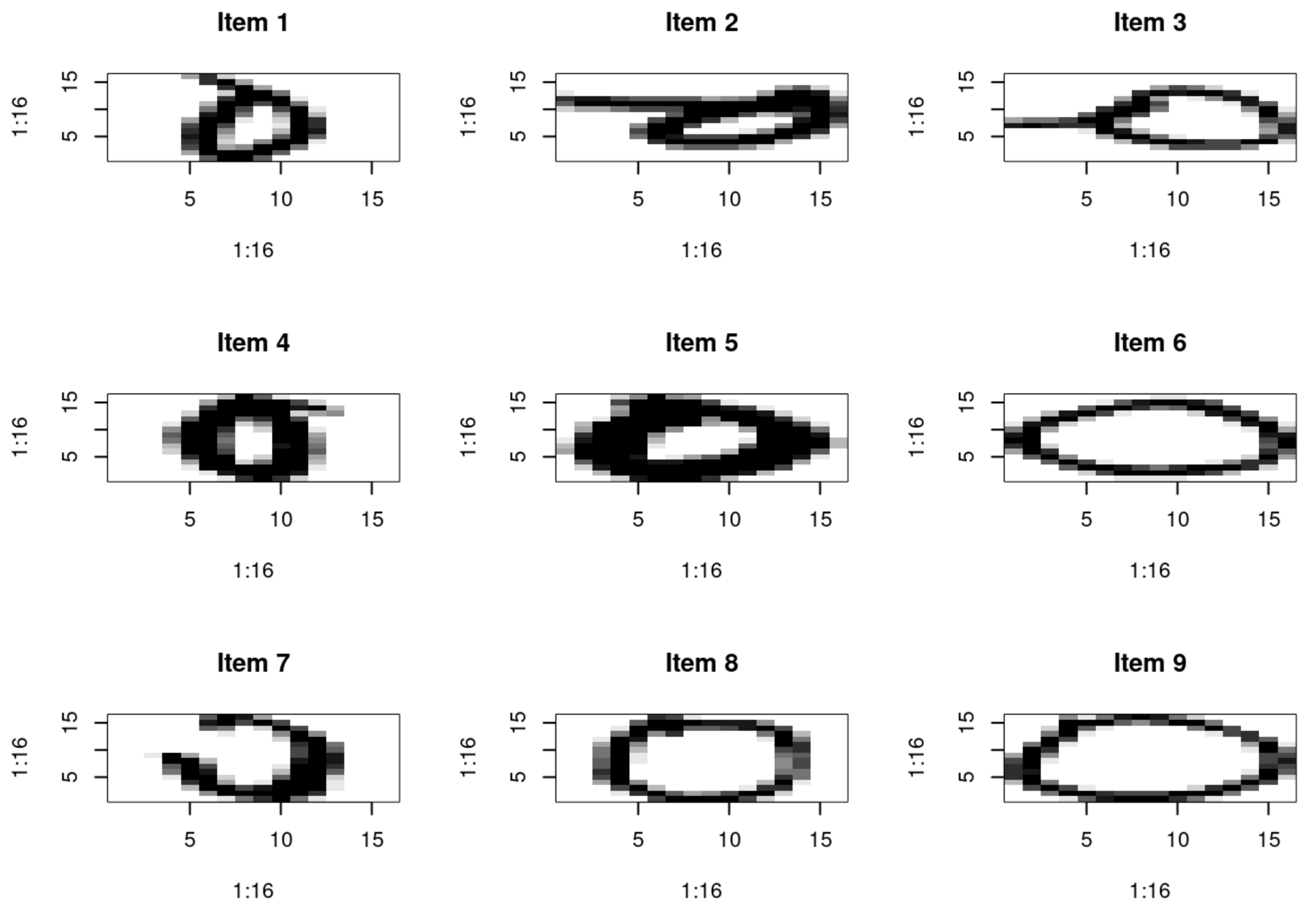


2b) Select points from scatterplot to cover variability

```
M <- as.data.frame(data.matrix(L[[1]]))
pts.var <- points.for.variability(M)
ids <- pts.var[3,]
aux.plot <- plot(L,n.col=3,main="Output of ISOMAP Algorithm")
points(M[ids,1],M[ids,2],col='yellow',pch=19,cex=4)
text (M[ids,1],M[ids,2],1:length(M[ids,1]), cex=2, font=2,col='red')
points(aux.plot,"sites",pch=19,cex=.6)
```



```
op <- par(mfrow=c(3,3))
draw.digits(ids,M,  pts.var[1,],  pts.var[2,])
```



```
par(op)
```

In ISOMAP, when we move from left to right in x-axis, the shape of the zeros are becoming thinner and wider, for example item 3 has wider and thinner shape compare to item 1. when we move upwards in y-axis, tail in zeros will be disappeared. For example item 7, 8 and 9 do not have any tail in their shape. Therefore, we expect that item 9 does not have a tail and has a thinner and wider shape compare to item 1.

2d) Compare your results using ISOMAP with those obtained using Local MDS

Looking at our results, we can see that our results are spread across the dimensions differently. As we know ISOMAP preserves geodesic distance while Metric MDS preserves Euclidean distance. Our goals is to reduce the dimensionality of the dataset with the least loss, so we expect that using geodesic distances produce effective results. In this example we saw that in ISOMAP we could describe one more feature, having tail in digits in addition to thikness or width.

3) Selecting the tuning parameters for ZERO digits

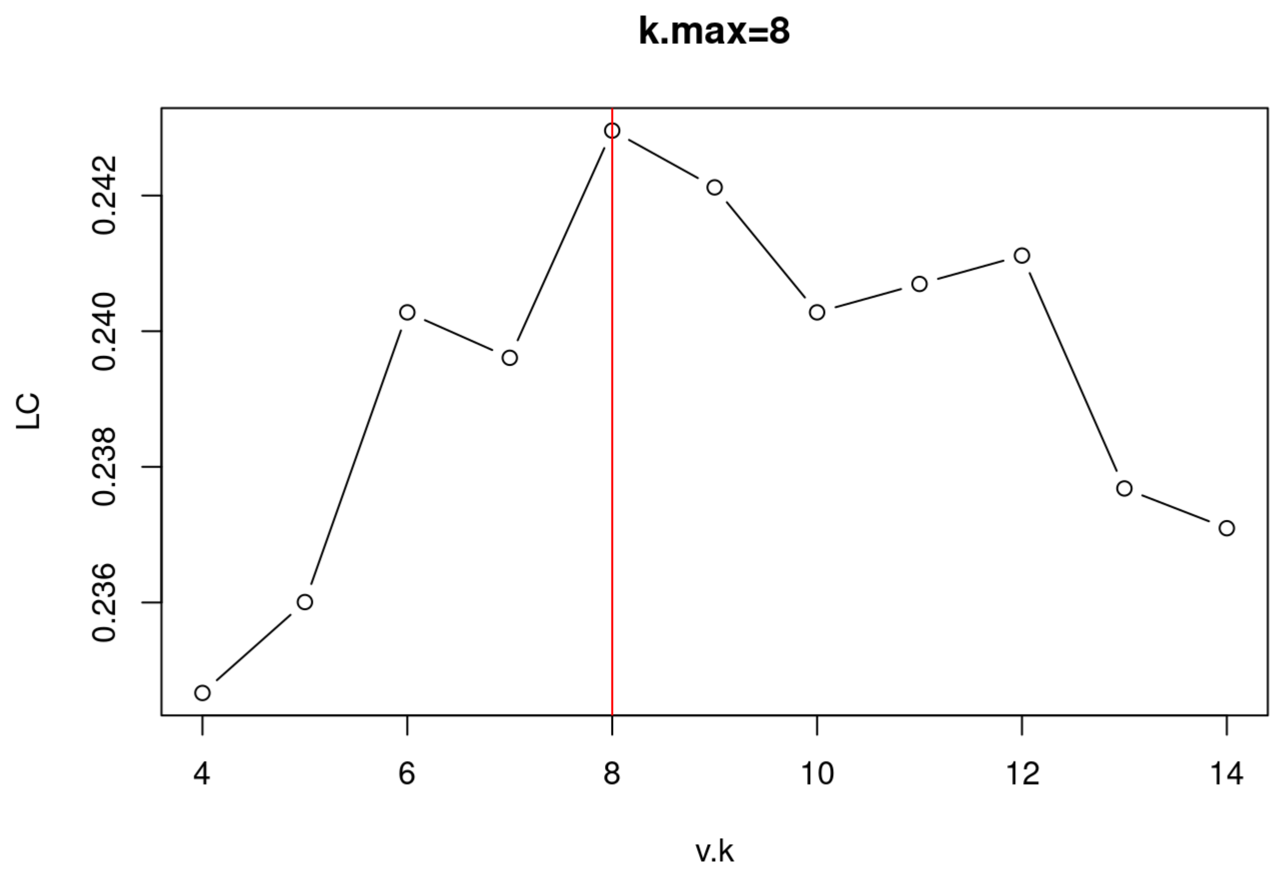
3a) Tune parameters using local continuity meta criteria

Use the local continuity meta criteria to select the tuning parameter k in ISOMAP for ZERO digits. Then describe graphically the low dimensional configuration corresponding to the optimal parameter.

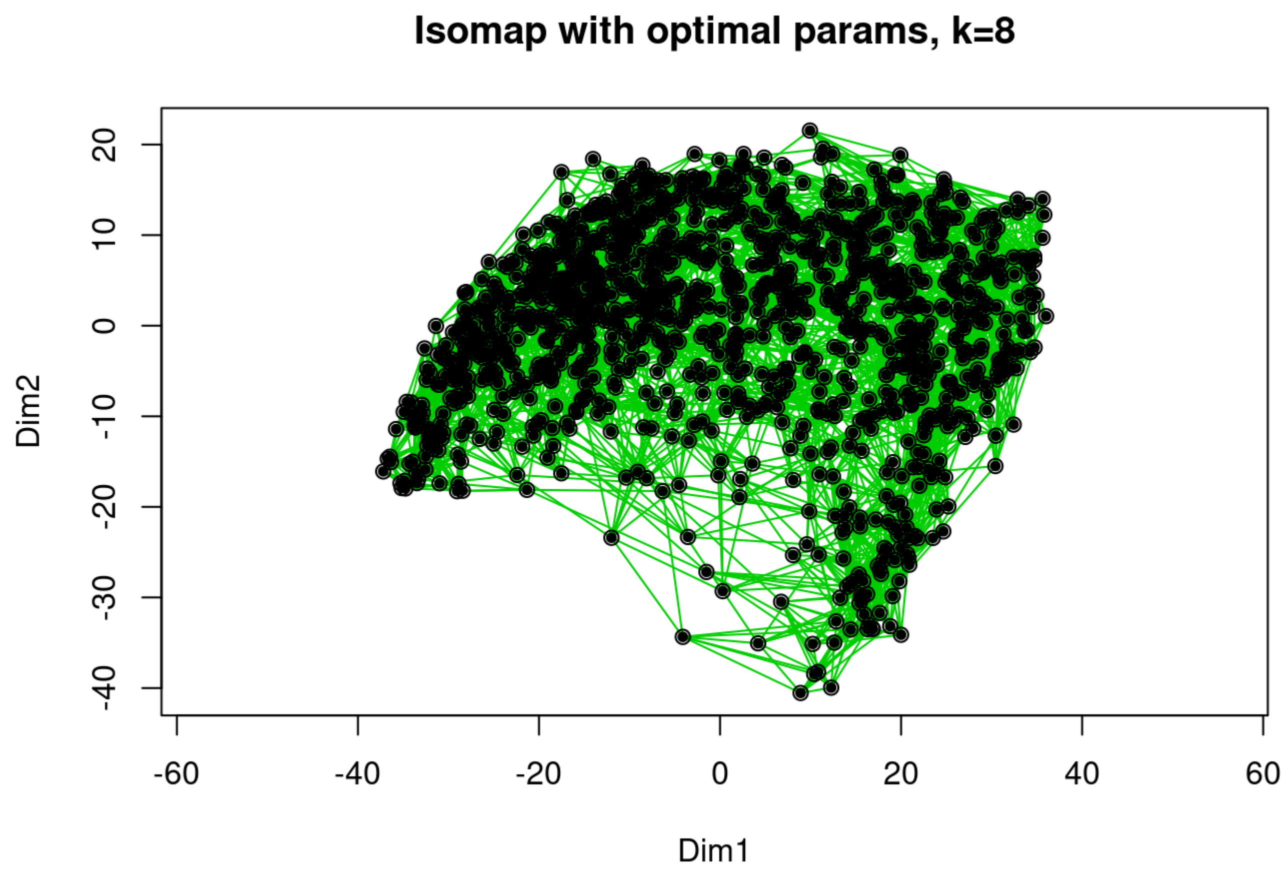
```r
LCMC <- function(D1,D2,Kp){
  D1 <- as.matrix(D1)
  D2 <- as.matrix(D2)
  n <- dim(D1)[1]
  N.Kp.i <- numeric(n)

  for (i in 1:n){
    N1.i <- sort.int(D1[i,],index.return = TRUE)$ix[1:Kp]
    N2.i <- sort.int(D2[i,],index.return = TRUE)$ix[1:Kp]
    N.Kp.i[i] <- length(intersect(N1.i, N2.i))
  }
  N.Kp <- mean(N.Kp.i)
  M.Kp.adj <- N.Kp/Kp - Kp/(n-1)

  return(list(N.Kp.i=N.Kp.i, M.Kp.adj=M.Kp.adj))
}
```

In k=8, we have the maximum value in LC. It means that intersection of the number of neighbors for point i in a high dimensional space with real distance and number of neighbors for point i in low dimensional space with euclidean distance is maximum in k=8. K started from 4, because values less than 4 give us fragmented graph. Large values increase the risk of short circuits.

```r
library(vegan)
v.k <- seq(4,14)
q <- 2
Kp <- 10
D1  <- dist(zip.train.0[,-1])
LC <- numeric(length(v.k))
ISOMAP.k <- vector("list",length(v.k))

for (i in 1:length(v.k)){
  ISOMAP.k[[i]] <- isomap(D1, ndim=q, k= v.k[i])
  D2.k <- dist(ISOMAP.k[[i]]$points[,1:q])
  LC[i] <- LCMC(D1,D2.k,Kp)$M.Kp.adj
}

i.max <- which.max(LC)
k.max <- v.k[i.max]
ISOMAP.max <- ISOMAP.k[[i.max]]

# plot the diffrenrt k baced on lc
plot(v.k, LC, type="b", main=paste0("k.max=",round(k.max,1)))
abline(v=k.max,col=2)
```



```
# plot the low configuration for optimal k
L <- isomap(D1, ndim=2, k=k.max)
aux.plot.opt <- plot(L,n.col=3,main=paste0("Isomap with optimal params, k=", k.max))
points(aux.plot.opt, "sites",pch=19,cex=.6)
```



3.b) b. (OPTIONAL)

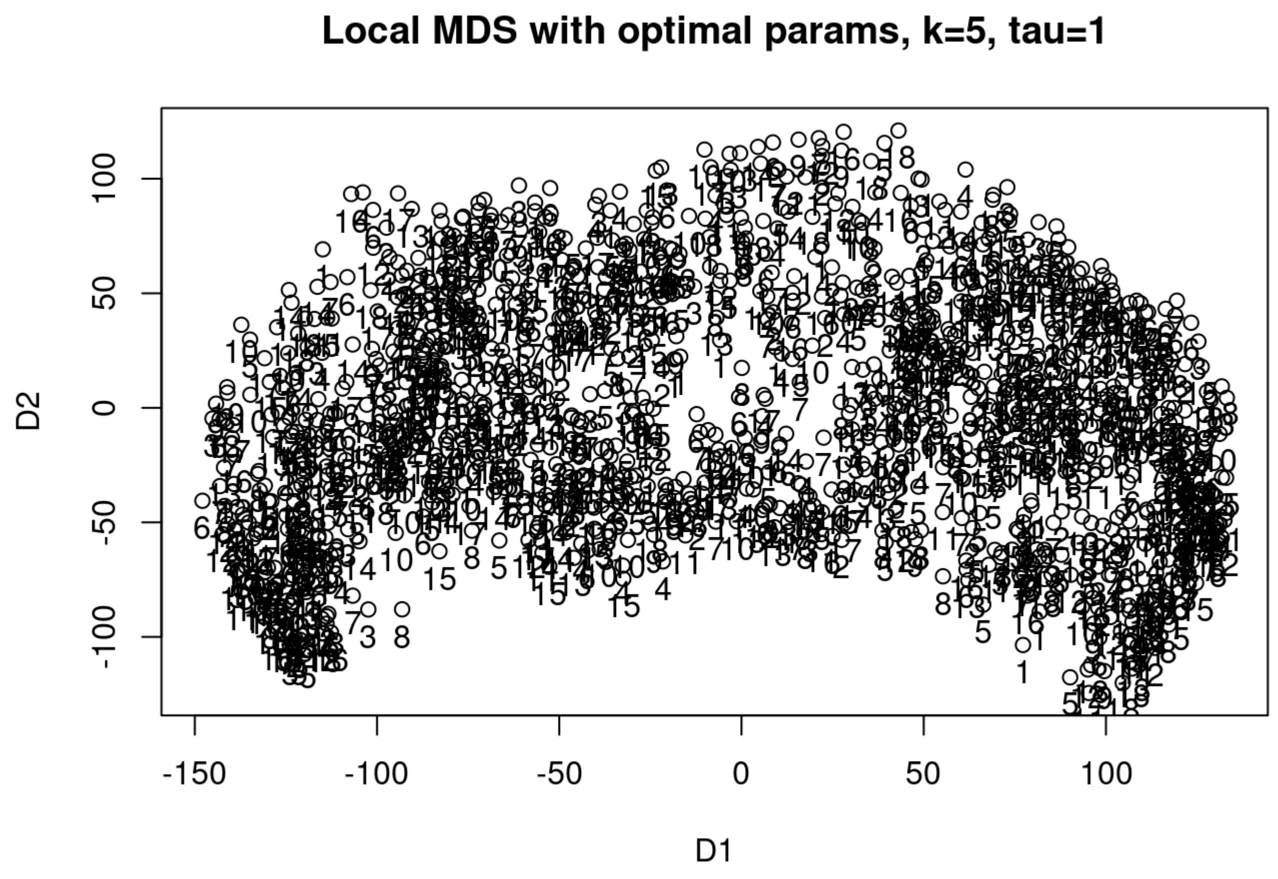
Use the local continuity meta criteria to select the tuning parameters k and τ in Local MDS for ZERO digits. Then describe graphically the low dimensional configuration corresponding to the optimal parameter.

```
dist.0 <- dist(zip.train.0)
n <- dim(dist.0)[1]

k <- 5
tau <- .05
q <- 2
conf0 <- cmdscale(dist.0, k=q)$points
lmds.S.res <- lmds(as.matrix(dist.0), init=conf0, ndim=q, k=k, tau=tau, itmax = 1000)
conf.lmds.S.res <- lmds.S.res$conf
```

```r
D1 <- dist(zip.train.0)
q <- 2
Kp <- 10
k <- c(5,10, 20)
tau <- c(.1,.5,1)

conf0 <- cmdscale(D1, k=q)$points
LC <- matrix(0,nrow=length(k),ncol=length(tau))
LocalMDS.k.tau <- array(vector("list",1),dim=dim(LC))

for (i in 1:length(k)){
  for (j in 1:length(tau)){
    LocalMDS.k.tau <-  lmds(as.matrix(D1), init=conf0, ndim=q, k=k[i], tau=tau[j], itmax
=20)
    D2.k.tau <- dist(LocalMDS.k.tau$conf)
    LC[i,j] <- LCMC(D1,D2.k.tau,Kp)$M.Kp.adj
  }
}

ij.max <- arrayInd(which.max(LC),.dim=dim(LC))
k.max <- k[ij.max[1]]
tau.max <- tau[ij.max[2]]
#LocalMDS.max <- LocalMDS.k.tau[[ij.max[1],ij.max[2]]]
print(paste0("k.max=",k.max,"; tau.max=",tau.max))
```

```
## [1] "k.max=5; tau.max=1"
```

```r
# plot low dim for optimal params
lmds.S.res <- lmds(as.matrix(D1), init=conf0, ndim=q, k=k.max, tau=tau.max, itmax=30)
conf.lmds.S.res <- lmds.S.res$conf
plot(conf.lmds.S.res, main=paste0("Local MDS with optimal params, k=", k.max, ", tau=",
 tau.max))
n <- D1[1]
text(conf.lmds.S.res[,1], conf.lmds.S.res[,2], 1:n, pos=1)
```


Local MDS with optimal params, k=5, tau=1
D2
D1
100
50
0
-50
-100
-150
-100
-50
0
50
100