Assignment: t-SNE

ZEROs in the ZIP numbers dataset

Mohana Fathollahi, Thanh Binh Viedena Vu

29 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)),...)
  invisible(
    if (!use.first){
      title(x[1])
    } else {
    }
  )
}
```

1) t-SNE for ZERO digits

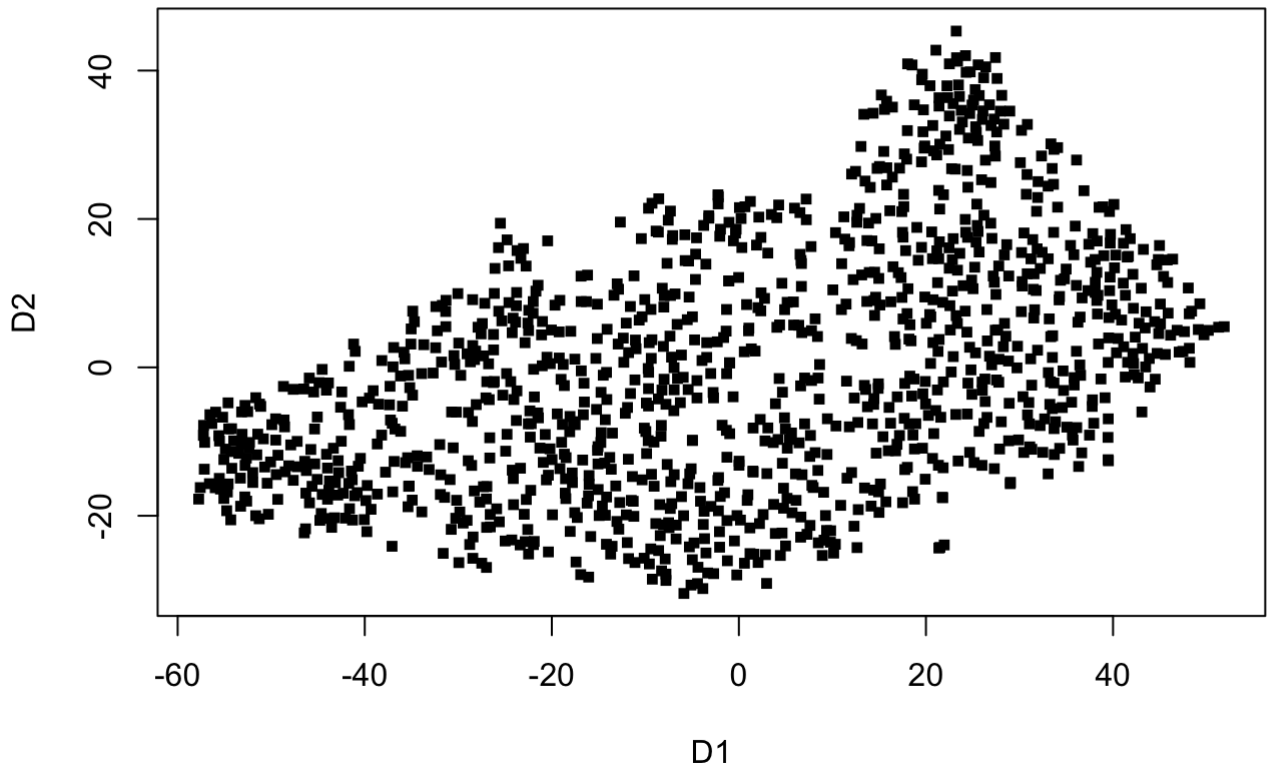
You must apply t-SNE to reduce the dimensionality of this dataset using the function Rtsne from package Rtsne.

```
library(Rtsne)
```

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

Look for a 2-dimensional (q = 2) configuration of the data using parameters perplexity= 40 and theta= 0 in Rtsne function. Do the scatterplot of the obtained 2-dimensional configuration.

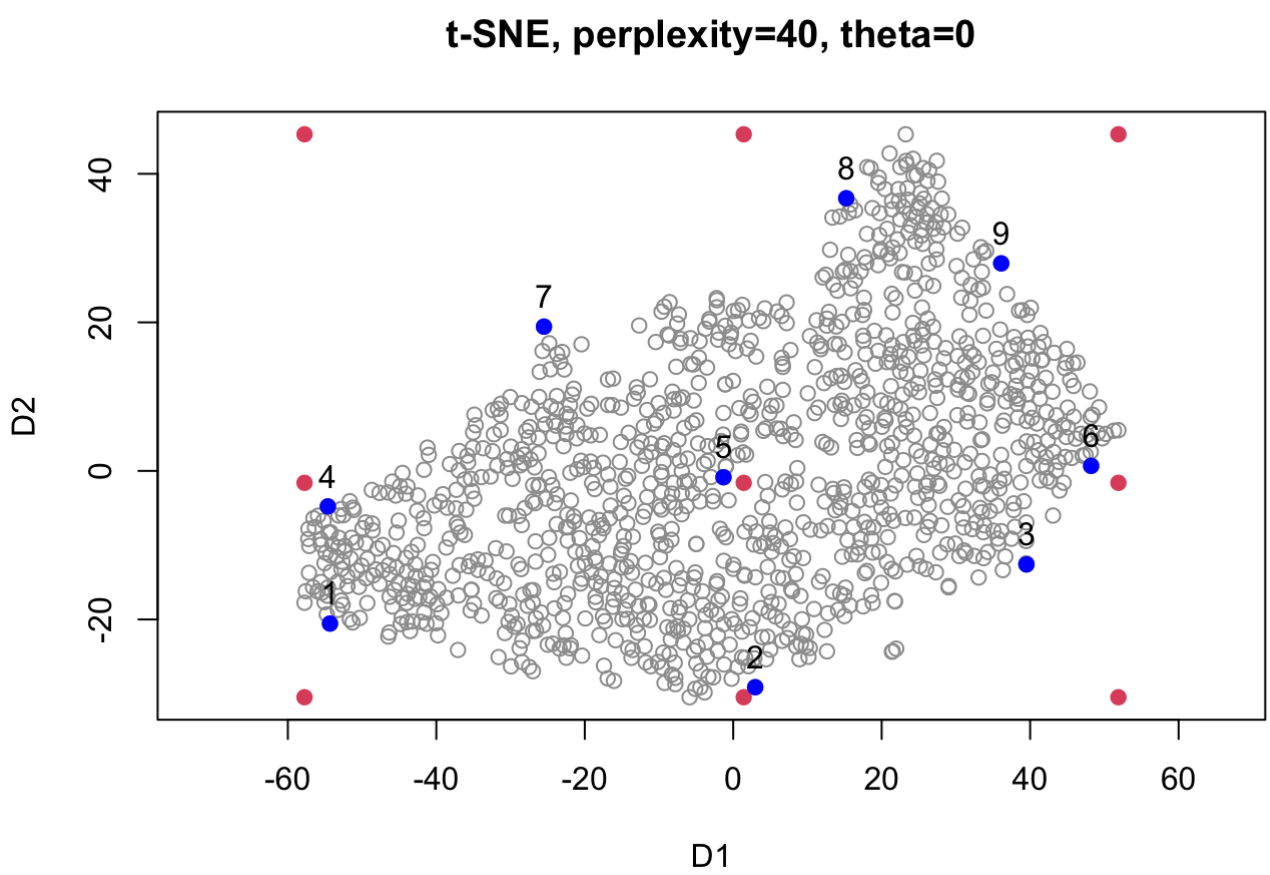
```
perplexity <- 40
theta <- 0

set.seed(42)
tsne_out <- Rtsne(zip.train.0, pca=FALSE, perplexity=perplexity, theta=theta)
plot(tsne_out$Y[,1:2],
     pch=c(15,17,19)[(zip.train.0[,1]%%3)+1],
     cex=.75, col=zip.train.0[,1]+1,
     xlab = "D1", ylab= "D2")
```



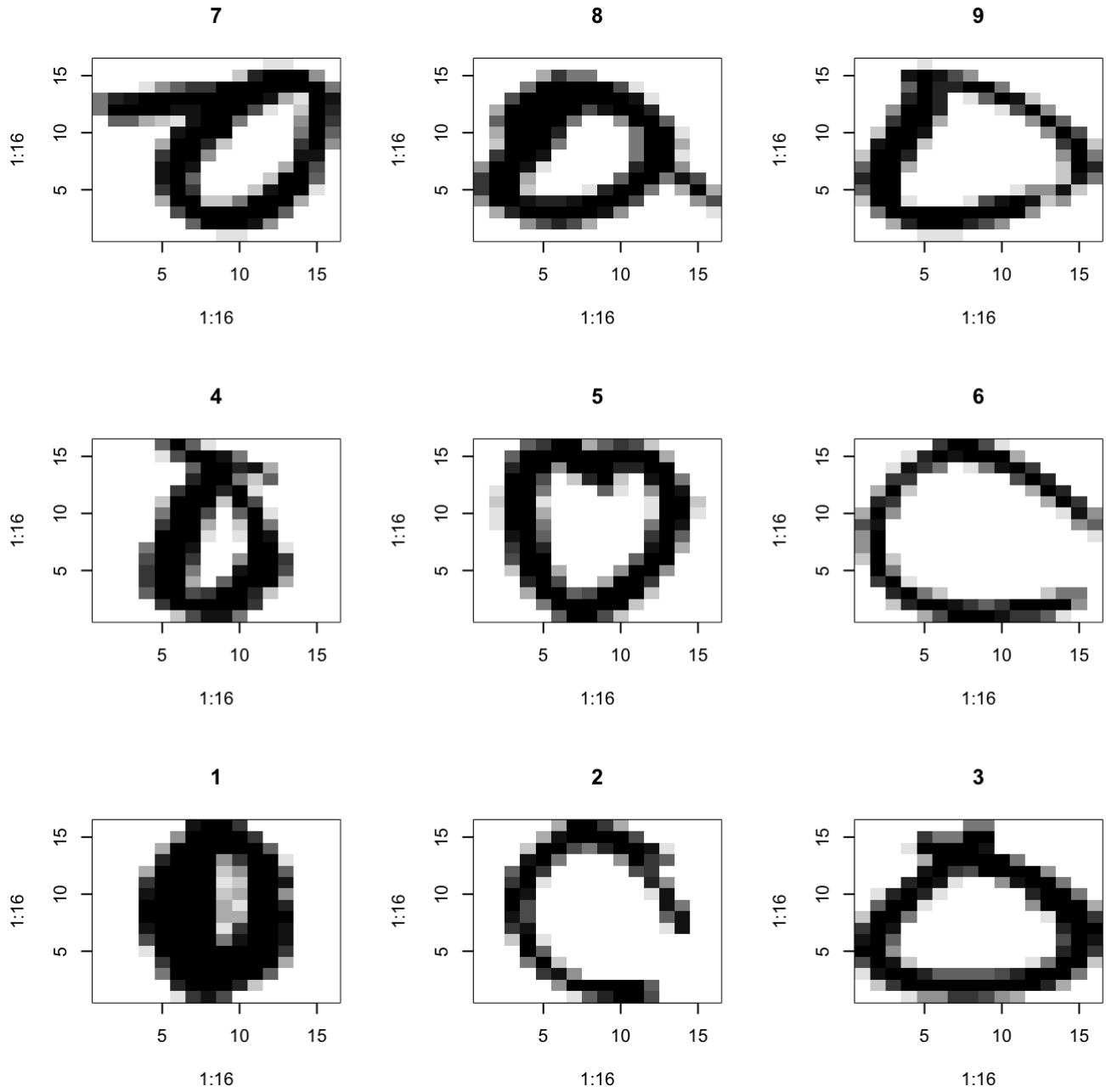
1b) Select points from scatterplot to cover variability

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 allows you to give an interpretation of the 2 coordinates obtained by t-SNE (observe how the shape of ZEROs changes when moving along each directions of the scatterplot).

```
library(plot3D)
xy<-mesh(boxplot.stats(tsne_out$Y[,1])$stats[c(1,3,5)],
      boxplot.stats(tsne_out$Y[,2])$stats[c(1,3,5)])

representers <- cbind(as.numeric(xy$x),as.numeric(xy$y))

d2.repr.tsne <- as.matrix(dist(rbind(representers,tsne_out$Y[,1:2])))[1:9,-(1:9)]

repr.obj <- apply(d2.repr.tsne,1,which.min)

plot(tsne_out$Y[,1:2],as=1,
     main=paste0("t-SNE, perplexity=",perplexity,", theta=",theta),col=8,
     xlab = "D1", ylab= "D2")

points(representers,col=2,pch=19)
points(tsne_out$Y[,1:2][repr.obj,],col="blue",pch=19)
text(tsne_out$Y[,1:2][repr.obj,1],tsne_out$Y[,1:2][repr.obj,2],1:9,pos=3)
```




```
op <- par(mfrow=c(3,3))
for (i in c(7,8,9,4,5,6,1,2,3)){
  plot.zip(zip.train.0[repr.obj[i],-1], use.first = TRUE, main=i)
}
```




```
par(op)
```

As we can see in this plot, when we move in x-axis from left to right shape of zeros are getting wider and thinner. When we move in y-axis to upside, zeros will have tail in one side.

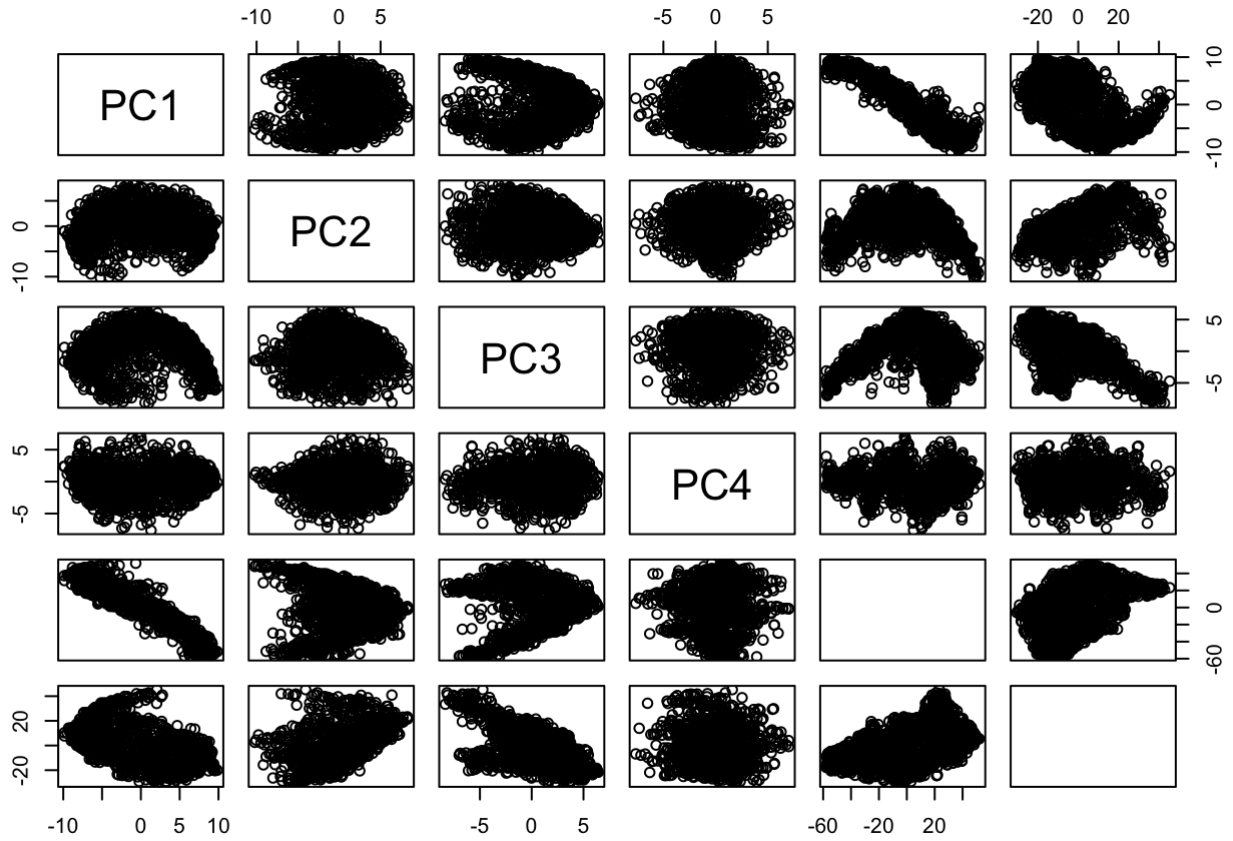
1c) Relate results from t-SNE to those from PC

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

```
PCA.zeros <- prcomp(zip.train.0[,-1], rank.=4)
summary(PCA.zeros)
```

```
## Importance of first k=4 (out of 256) components:
##                           PC1    PC2    PC3     PC4
## Standard deviation     5.417 3.6391 3.2379 2.37879
## Proportion of Variance 0.282 0.1273 0.1008 0.05438
## Cumulative Proportion  0.282 0.4093 0.5100 0.56440
```

```
pairs(cbind(PCA.zeros$x,tsne_out$Y[,1:2]))
```



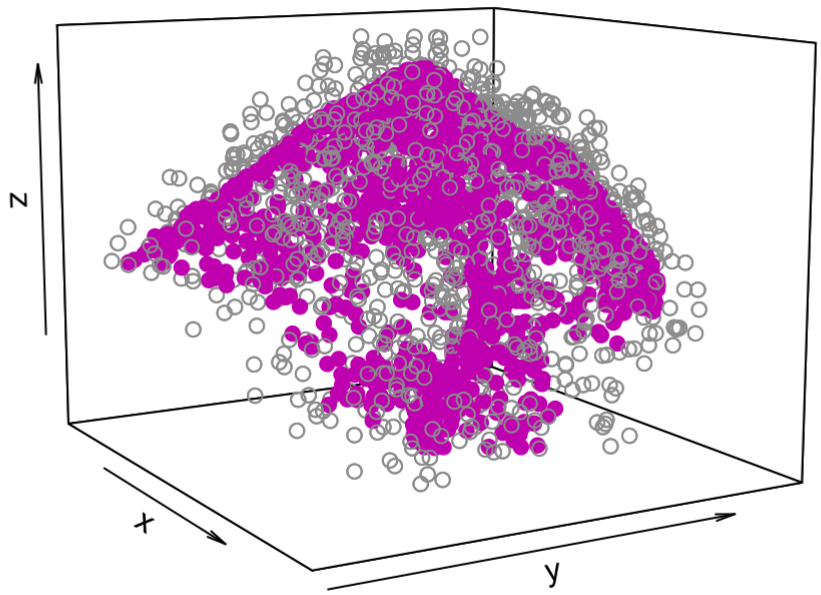

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
df.PCA.tsne <- as.data.frame(cbind(PCA.zeros$x,tsne_out$Y[,1:2]))
smooth.PC1 <- gam(PC1~s(V5,V6), data=df.PCA.tsne)$fitted.values
smooth.PC2 <- gam(PC2~s(V5,V6), data=df.PCA.tsne)$fitted.values
smooth.PC3 <- gam(PC3~s(V5,V6), data=df.PCA.tsne)$fitted.values
```

```
points3D(PCA.zeros$x[,1],PCA.zeros$x[,2],PCA.zeros$x[,3],col=8,
         colvar = NULL, phi = 10, theta = 60, r =sqrt(3), d = 3,scale=FALSE)
points3D(smooth.PC1,smooth.PC2,smooth.PC3,add=TRUE,col=6,pch=19)
```



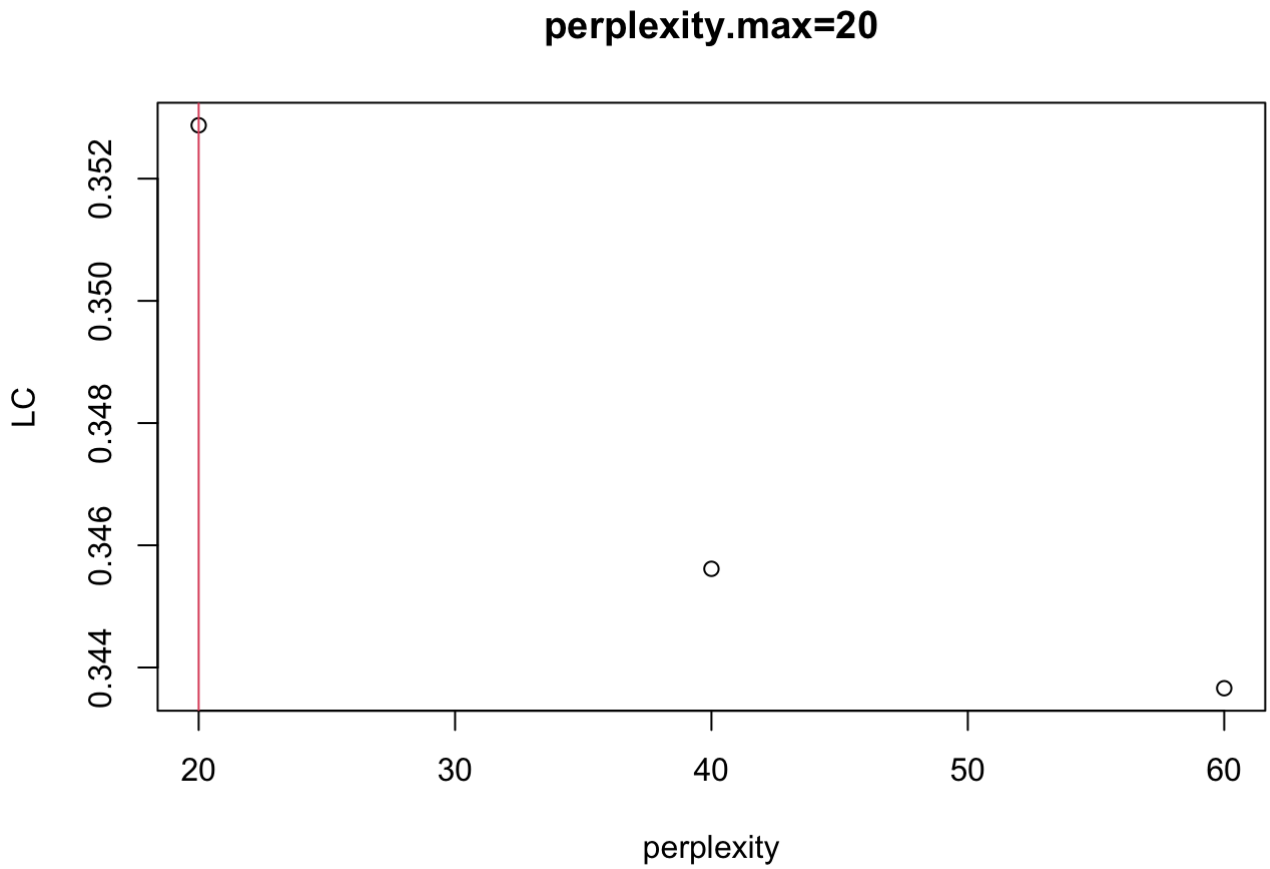

```
library(rgl)
points3d(PCA.zeros$x[,1],PCA.zeros$x[,2],PCA.zeros$x[,3],col=8,size=6)
points3d(smooth.PC1,smooth.PC2,smooth.PC3,add=TRUE,col=6,pch=19,size=10)
```

###3)(OPTIONAL) Selecting the tuning parameters for ZERO digits

a. Use the local continuity meta criteria to select the tuning parameter perplexity in t-SNE for ZERO digits (use q = 2 and theta= 0). Then describe graphically the low dimensional configuration corresponding to the optimal parameter. Indication: As tentative values for perplexity use c(20,40,60).

```
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))
}
```

```
require(Rtsne)
set.seed(4444)

D1 <- dist(zip.train.0[,-1])
q <- 2
Kp <- 3
theta <- 0
perplexity <- c(20,40,60)

LC <- numeric(length(perplexity))
Rtsne.k <- vector("list",length(perplexity))

for (i in 1:length(perplexity)){
    Rtsne.k[[i]] <- Rtsne(D1, perplexity=perplexity[i], dims=q,
                          theta=0, pca=FALSE, max_iter = 100)

    D2.k <- dist(Rtsne.k[[i]]$Y)   #distance in low dimension
    LC[i] <- LCMC(D1,D2.k,Kp)$M.Kp.adj
    #print(c(i,j,LC[i,j]))
}

i.max <- which.max(LC)
perplexity.max <- perplexity[i.max[1]]
Rtsne.max <- Rtsne.k[[i.max]]

plot(perplexity,LC, main=paste0("perplexity.max=",perplexity.max))
abline(v=perplexity[i.max],col=2)
```



```
tsne_out_2 <- Rtsne(zip.train.0, pca=FALSE, perplexity=perplexity.max, theta=theta)
plot(tsne_out_2$Y[,1:2],
     pch=c(15,17,19)[(zip.train.0[,1]%%3)+1],
     cex=.75, col=zip.train.0[,1]+1,
     xlab = "D1", ylab= "D2", main=paste0("tSNE with optimal param,perplexity =", per
plexity.max))
```