|  |
| --- |
| setwd('c:/users/lux/Desktop/ML/P2')  #Data load  dat<-read.csv("c:/users/lux/Desktop/ML/P2/mnist.csv",header = FALSE)  length(dat)#785 with first column  #Removing first column  #dat<-dat[,-1]  length(dat) #784  #Question 1 : Employ either the princomp or prcomp function to perform PCA on the  #784 columns of the data set which contain the grey scale pixel information.  #Do not include the first column as this column merely indicates the digit  #0-9 that a given row encodes.  #Princomp function to perform pca on the data  #mnist\_data=dat[,-(which((1:784)%%28<=2|(1:784)%%28>=26|1:784%/%28<=2|1:784%/%28>=26)+1)]  # only 530 variables include grey data .  prin<-princomp(dat[-1])  summary(prin)  #Report the number of principal components needed to account for 98% ofthe variance of the original data set.  PoV <- prin$sdev^2/sum(prin$sdev^2)  var<-data.frame(PoV[1:261])  #Question 3  #Contribution of all 261 components leading to 98%  write.csv(var,'pcaper.csv')  View(var)  #261 components contribute to 98%  sum(PoV[1:261])  cumsum(prin$sdev^2 / sum(prin$sdev^2))  loading<-prin$loadings[1:261]  loading  #bonus with 28\*28 grid fo the original dataset for first 10 components  bdata<-dat[-1]  pc<-princomp(bdata)  step2 <- scale(bdata[,(1:ncol(bdata))], pc$center, pc$scale) %\*% pc$loadings  data\_table <- data.table(bdata$V2,step2[,1:10]%\*%t(pc$loadings)[1:10,])  par(mfrow=c(10,10),mar=c(0.1,0.1,0.1,0.1))  #Plotting 100 observations  for (i in 1:100)  {  new\_mtrix <- matrix(as.numeric(data\_table[i,784:2,with=F]),nrow = 28,ncol=28,byrow = F)  new\_mtrix <-new\_mtrix[nrow(new\_mtrix):1,]  image(new\_mtrix)  }  number <- read.csv("c:/users/lux/Desktop/ML/P2/mnist.csv",  header = F, sep = ",")  number\_2 <- number[which(number[,1]==2),]  dim(number\_2)  number\_mat <- matrix(as.numeric(number\_2[3,-1]), 28, 28, byrow = T)  mat1 <- apply(number\_mat, 2, rev)  image(1:28, 1:28, t(mat1), axes = FALSE, xlab = "", ylab = "",  col = grey(seq(1, 0, length = 255)))  number\_ <- number\_2[,which(!apply(number\_2,2,FUN = function(x){all(x == 0)}))]  number\_$V1 <- NULL  dim(number\_)  out\_pca <- princomp(number\_, cor = FALSE)  # loadings of the first four PCs  loadings\_pca <- cbind(round(out\_pca$loadings[,1], digits = 4),  round(out\_pca$loadings[,2], digits = 4),  round(out\_pca$loadings[,3], digits = 4),  round(out\_pca$loadings[,4], digits = 4))  write.csv(loadings\_pca,'loadings.csv')  # proportion of explained variance  out\_pca$sdev["Comp.1"]^2/sum(out\_pca$sdev^2)  used\_pca <-which(names(number)%in%c(colnames(number\_)))  out\_pca$sdev["Comp.2"]^2/sum(out\_pca$sdev^2)  used\_pca <-which(names(number)%in%c(colnames(number\_)))  out\_pca$sdev["Comp.3"]^2/sum(out\_pca$sdev^2)  used\_pca <-which(names(number)%in%c(colnames(number\_)))  out\_pca$sdev["Comp.4"]^2/sum(out\_pca$sdev^2)  used\_pca <-which(names(number)%in%c(colnames(number\_)))  pc <- matrix(0, nrow = 4, ncol = 784)  v <-1  for (i in used\_pca) {  pc[,i] <- loadings\_pca[v,]  v <- v+1  }  number\_mat <- matrix(as.numeric(number\_2[3,-1]), 28, 28, byrow = T)  mat1 <- apply(number\_mat, 2, rev)  col <- colorRampPalette(c("white", "blue"))  col2 <- colorRampPalette(c("white", "yellow"))  for (i in 1:4) {    pca\_mat <- matrix(as.numeric(pc[i,]), 28, 28, byrow = T)  mat\_pca <- apply(pca\_mat, 2, rev)      #blue: positive loadings, yellow: negative loadings)  A <- t(mat1)  B <- (A>1) \* t(mat\_pca)  B[ B<=0.0 ] <- NA    image(1:28, 1:28, A, axes = FALSE, xlab = "", ylab = "", col = grey(seq(1, 0, length = 255)))  image(1:28, 1:28, B, axes = FALSE, xlab = "", ylab = "",  col= col(10), add=T)    A <- t(mat1)  B <- (A>1) \* t(mat\_pca)  B[ B>=0.0 ] <- NA    image(1:28, 1:28, B, axes = FALSE, xlab = "", ylab = "",  col= col2(10), add=T)    B <- t(mat\_pca)  B[A>1] <-NA  B[ B>=0.00 ] <- NA  image(1:28, 1:28, B, axes = FALSE, xlab = "", ylab = "",  col= col(10), add=T) # overlay    B <- t(mat\_pca)  B[A>1] <-NA  B[ B<=0.00 ] <- NA  image(1:28, 1:28, B, axes = FALSE, xlab = "", ylab = "",  col= col2(10), add=T) # overlay    } |