Kevin Gong

kg2445

HW02

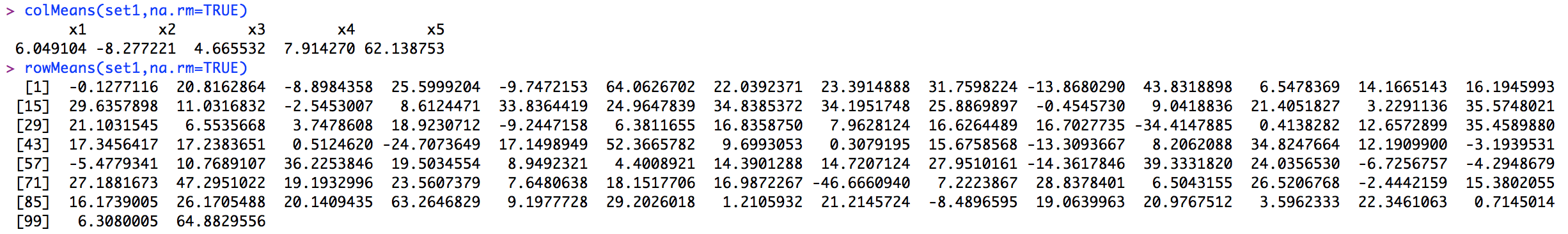
Stat W4240

Section 2

**Homework 2**

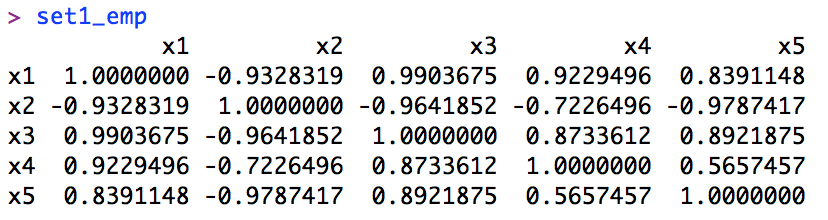
**Problem #1**

Part a)



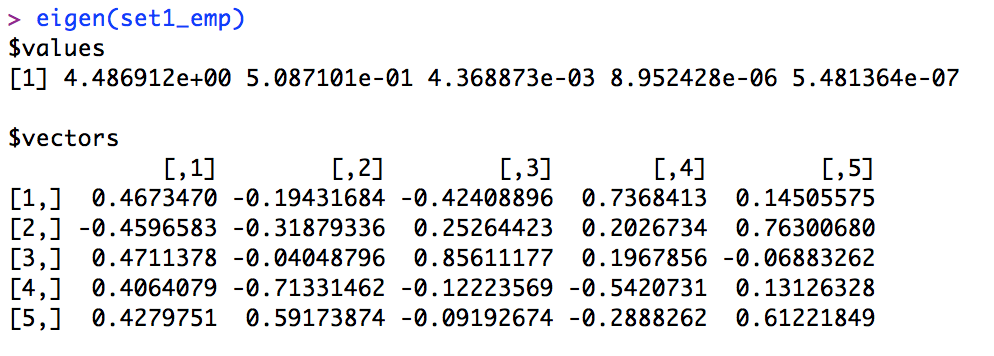
This matrix has 5 columns and 100 rows, and we can see that the column and row means vary in both magnitude and sign. The second column has a negative mean, while the fifth column’s mean is significantly larger than those of the other columns. Among the rows, there are more positive row means than negative ones.

Part b)



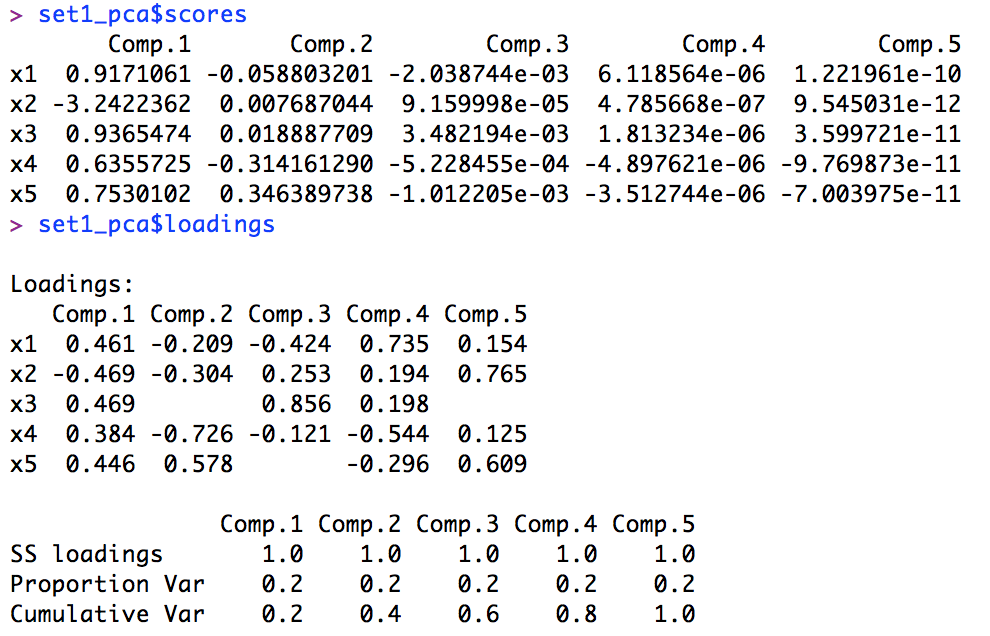
The diagonal values of the covariance matrix are all 1, while the off-diagonal values are symmetric around the main diagonal. Thus, we know that the covariance matrix is symmetric and standardized.

Part c)

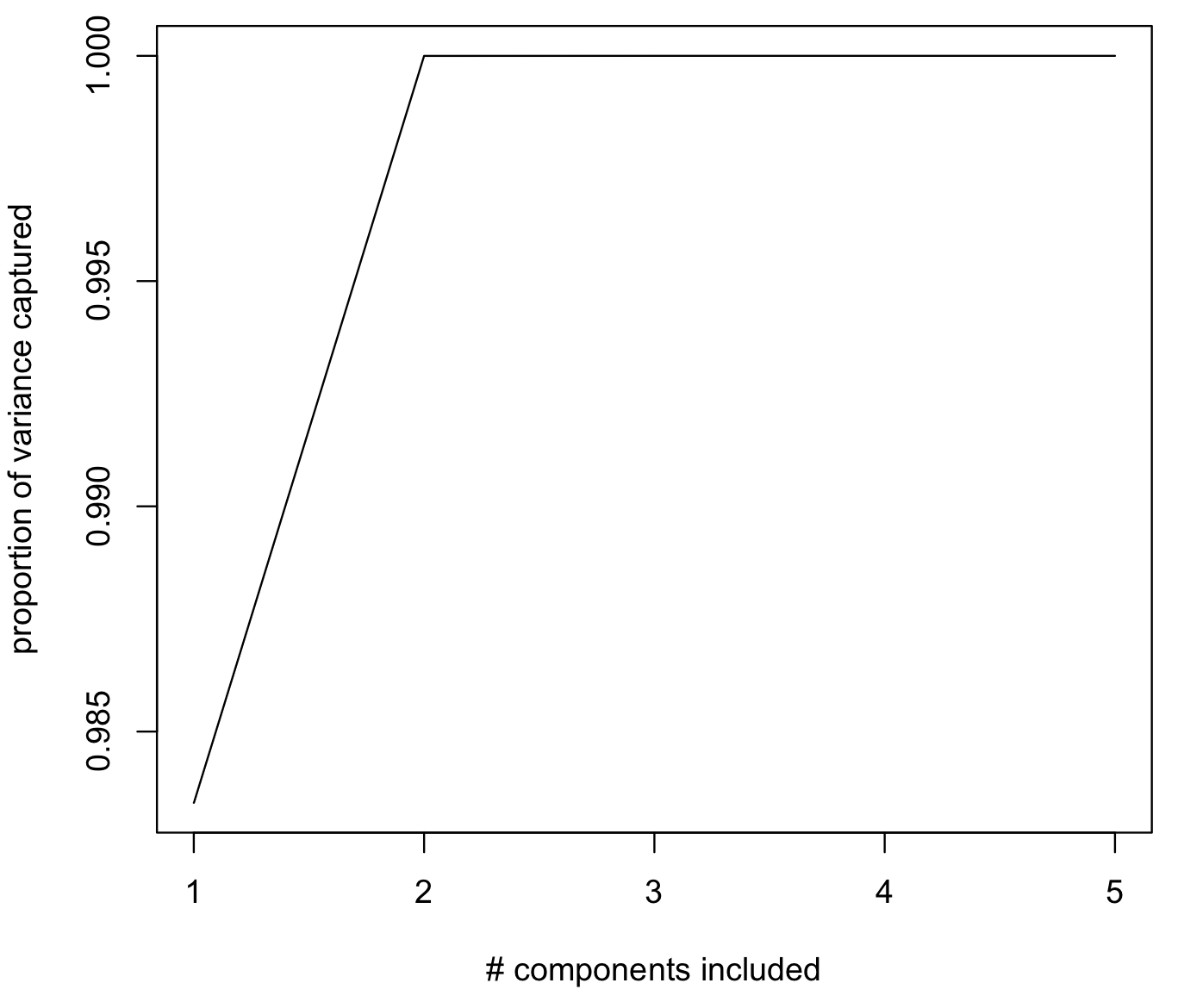


We know that the left eignenvectors are transposes of the right eigenvectors of the transposed covariance matrix. In this case, our left and right eigenvectors are the same since our covariance matrix is symmetric.

Part d)

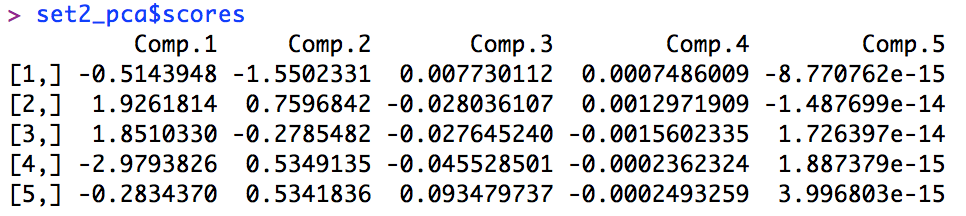


Part e)



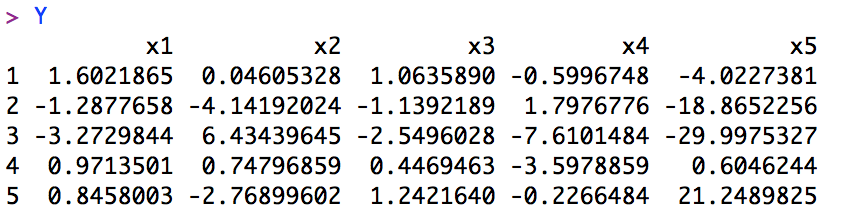
We should include 2 components, as additional components beyond the first two contribute very insignificantly toward the proportion of variance captured.

Part f)

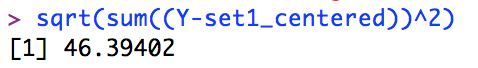


Part g)

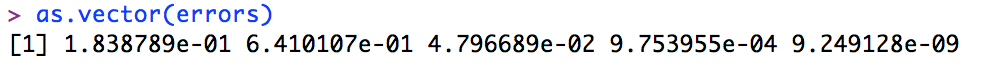
Coordinates of the projection in the original space:



Euclidean distance:



Part h)



The errors are all positive, though very small, for the 5 new points. The direction of the error for the 5 new points is likely perpendicular to the direction of the projections in the original space since this generates the smallest possible error.

**Problem #2**

Part a)



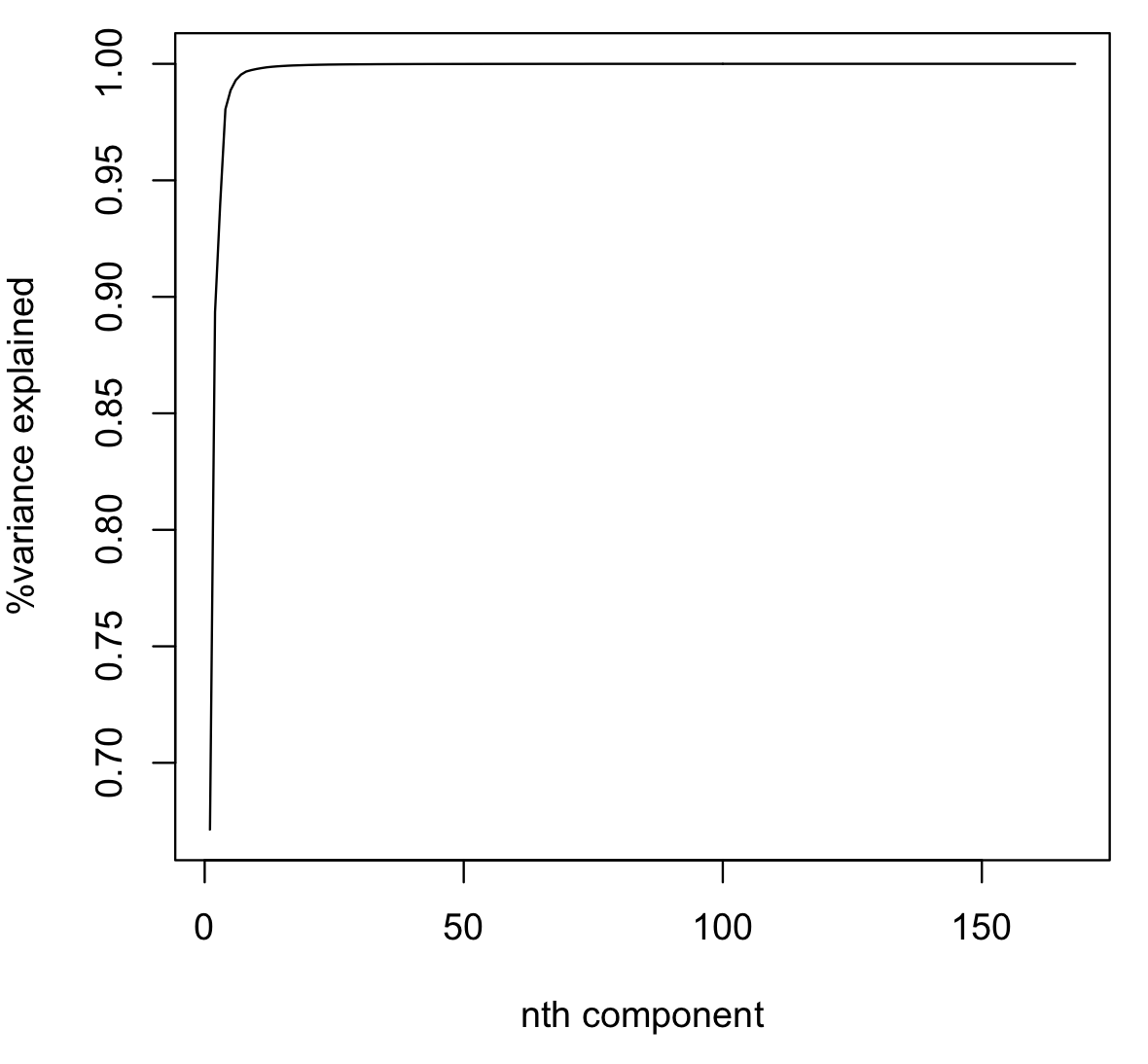
This matrix is 152 rows by 32,256 columns, where each row represents a different picture.

Part b)

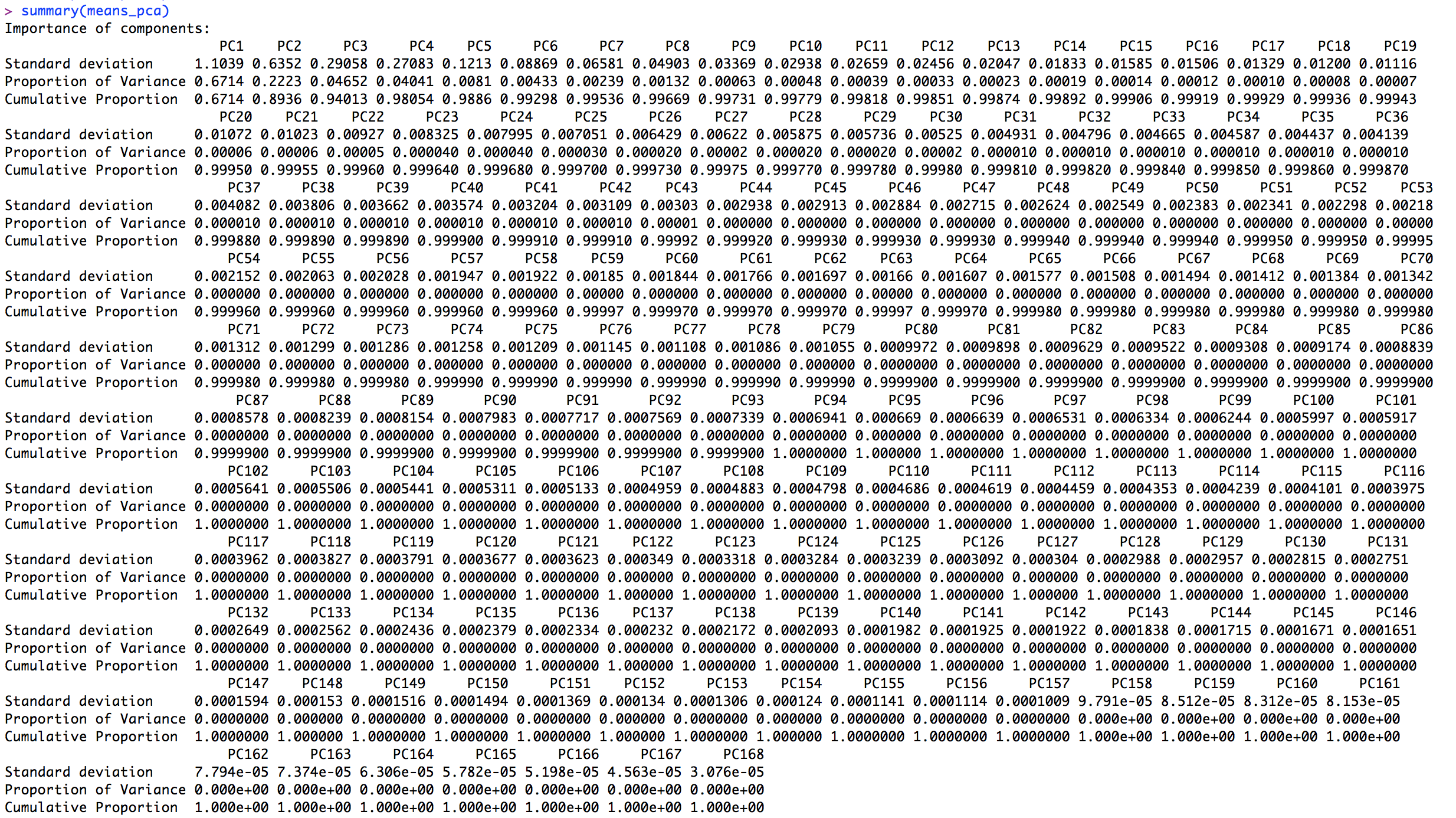


Part c)

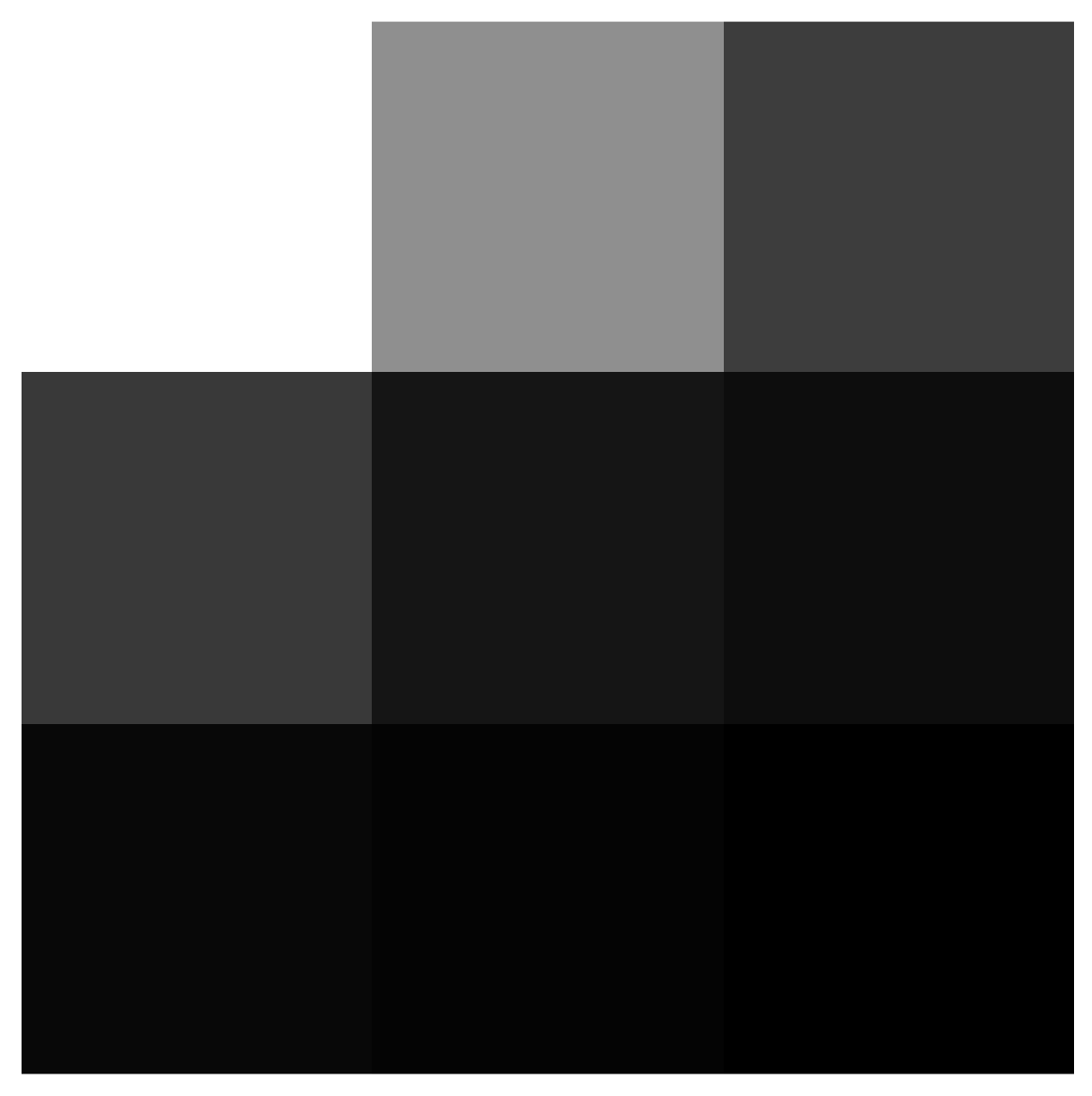
Number of components vs. proportion of variance explained:



Principal components:

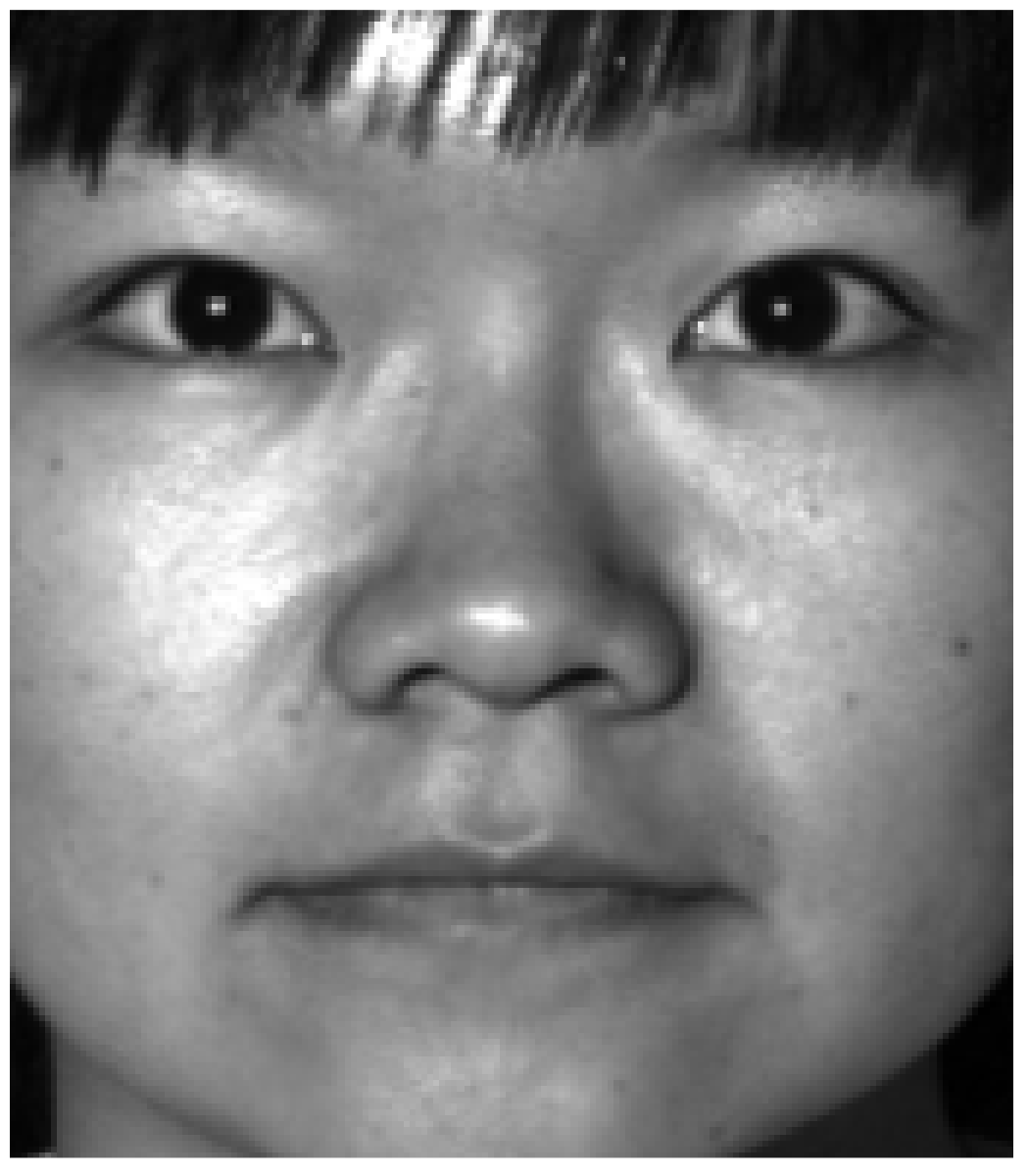
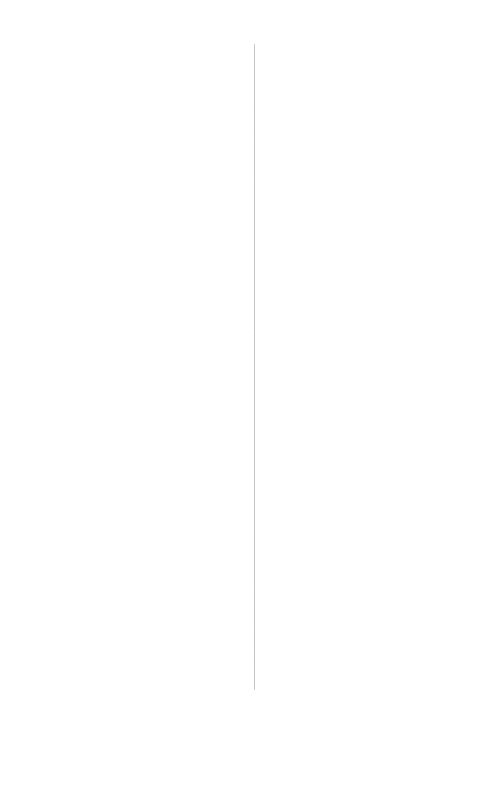


Part d)



The eigenfaces take on the standard ranges of pixmap class objects of 0 to 1, where 1 represents white and 0 represents black. These eigenfaces represent eigenvectors of our covariance matrix of faces, or more generally the principal components of a distribution of faces. We generated our principal components in part c, and these form a basis for the images of all of the faces.

Part e)

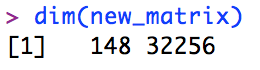


Unfortunately, we failed to generate the 5x5 plot, as our code only produced an image of a vertical grey line (see middle image above). However, we understand that by adding eigenfaces in the following manner…

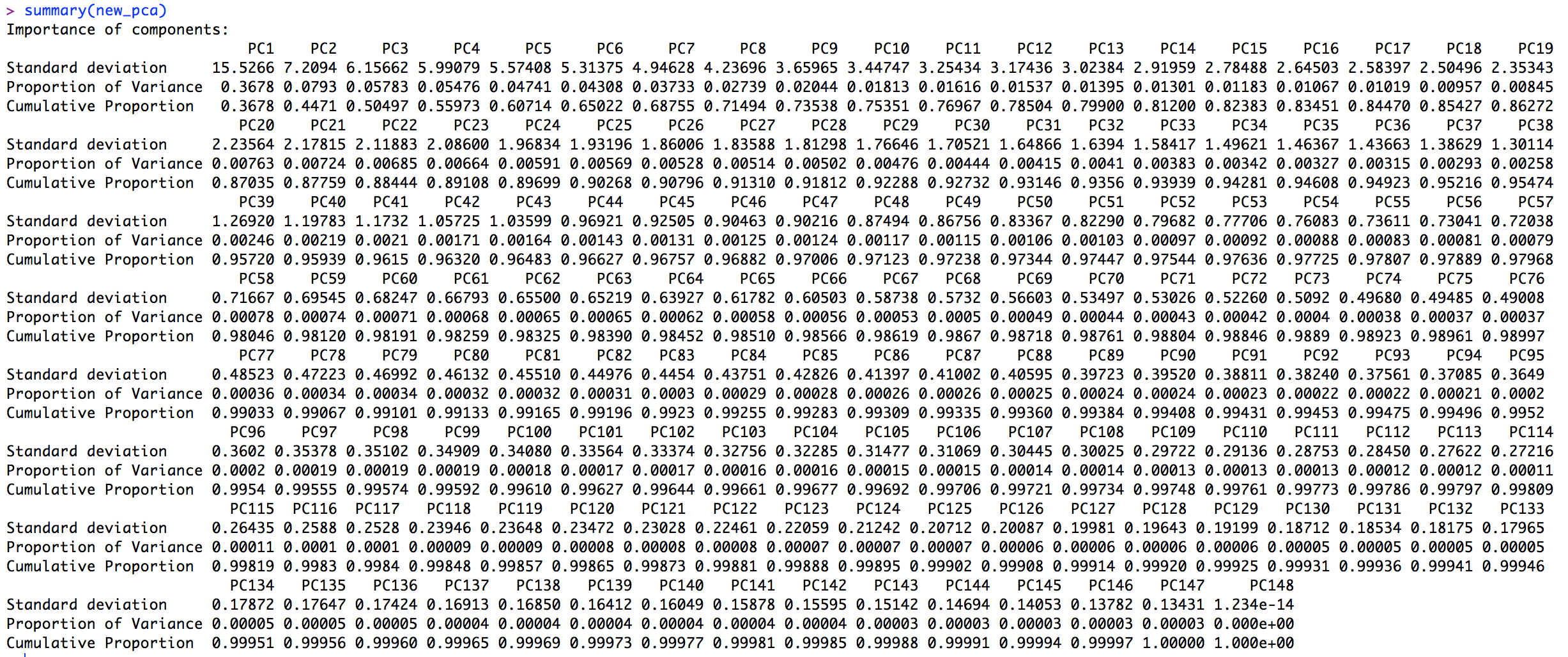
mean\_face + score[1]\*loading[1] + ... + score[m]\*loading[m]

…we slowly transforms our mean face into the face of Subject 5. It would likely take us a fairly substantial number of eigenvectors (at least about 80~100) for us to be able to recognize the person.

Part f)



Removing subject 1 leaves us with a new matrix of faces of dimension 148 by 32256. We also successfully run prcomp() to obtain new principal components (see below).



Unfortunately, we were unable to generate a reconstructed image, though we understand that we are using the eigenfaces of the remaining images to reconstruct subject 1’s image. We know that the reconstructed image will look somewhat like the original image since we are eliminating data on the other subjects’ faces from the mean face which thus allows us to approximate the image of subject 1.

**Code**

#############################

# Kevin Gong

# STAT W4240

# Homework 2 , Problem 1

# 2/19/14

#

#############################

#################

# Setup

#################

# make sure R is in the proper working directory

# note that this will be a different path for every machine

setwd("~/Dropbox/SIPA/Data Mining/HW2")

# first include the relevant libraries

# note that a loading error might mean that you have to

# install the package into your R distribution. From the

# command line, type install.packages("pixmap")

library(pdist)

#################

# Problem 1a

#################

set1 = read.csv(file="hw02\_q1\_p1.csv")

head(set1)

colMeans(set1,na.rm=TRUE)

rowMeans(set1,na.rm=TRUE)

#################

# Problem 1b

#################

set1\_centered = scale(set1, center=TRUE)

set1\_emp = cov(set1\_centered)

set1\_emp

#################

# Problem 1c

#################

eigen(set1\_emp)

#################

# Problem 1d

#################

set1\_pca = princomp(set1\_emp)

set1\_pca$scores

set1\_pca$loadings

#################

# Problem 1e ??

#################

set1\_pca$sdev^2/sum(set1\_pca$sdev^2)

cumsum(set1\_pca$sdev^2/sum(set1\_pca$sdev^2))

plot(cumsum(set1\_pca$sdev^2/sum(set1\_pca$sdev^2)),type='l',xlab="# components included", ylab="proportion of variance captured")

#################

# Problem 1f

#################

set2 = read.csv(file="hw02\_q1\_p2.csv")

set2\_centered = scale(set2, center=TRUE)

set2\_pca = princomp(set2\_centered)

set2\_pca$scores

#################

# Problem 1g

#################

W = set1\_pca$loadings[,1:2]

X = set2\_centered[1:5,1:5]

Y = X\*t(W)

#euclidean distances

sqrt(sum((Y-set1\_centered))^2)

pdist(Y,set1\_centered)

dist(Y,set1\_centered)

#################

# Problem 1h

#################

errors = set2\_pca$sdev-set1\_pca$sdev

as.vector(errors)

#############################

# Kevin Gong

# STAT W4240

# Homework 2 , Problem 2

# 2/19/14

#

#############################

#################

# Setup

#################

# make sure R is in the proper working directory

# note that this will be a different path for every machine

setwd("~/Dropbox/SIPA/Data Mining/HW2")

# first include the relevant libraries

# note that a loading error might mean that you have to

# install the package into your R distribution. From the

# command line, type install.packages("pixmap")

library(pixmap)

#################

# Problem 2a

#################

# the list of pictures

dir\_list\_1 = dir(path="CroppedYale/",all.files=FALSE)

view\_list = c( 'P00A+000E+00' , 'P00A+005E+10' , 'P00A+005E-10','P00A+010E+00')

# preallocate an empty list

this\_face\_row = vector()

# initialize an empty matrix of faces data

faces\_matrix = vector()

pic\_list = c( 01,02,03,04,05,06,07,08,09,10,11,12,13,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39)

#generating a matrix of photos of the 4 desired views

for (i in 1:38) {

placeholder\_matrix <- NULL

for (j in 1:4) {

filename= sprintf("CroppedYale/%s/%s\_%s.pgm",dir\_list\_1[i],dir\_list\_1[i],view\_list[j])

print(filename)

face=read.pnm(file=filename)

pic\_data <-getChannels(face)

pic\_data\_vec <- as.vector(pic\_data)

placeholder\_matrix=rbind(placeholder\_matrix,pic\_data\_vec)

}

faces\_matrix=rbind(faces\_matrix,placeholder\_matrix)

}

dim(faces\_matrix)

#################

# Problem 2b

#################

#computing the mean of the faces

faces.matrix\_v2=as.matrix(faces\_matrix)

means\_col=colMeans(faces.matrix\_v2)

means\_mat=as.matrix(means\_col)

#reshape this matrix into 192x168

means\_mat\_ref=matrix(means\_mat,192,168)

#double check dimensions to make sure it worked

dim(means\_mat\_ref)

#generate image of mean face

means\_face=pixmapGrey(means\_mat\_ref)

plot(means\_face)

#save image of mean face

filename = 'hw02\_02b.png'

dev.copy(device=png, file=filename, height=600, width=800)

dev.off()

#################

# Problem 2c

#################

means\_pca = prcomp(means\_mat\_ref)

summary(means\_pca)

plot(summary(means\_pca)$importance[3,], type="l", ylab="%variance explained", xlab="nth component")

means\_pca$x

#################

# Problem 2d

#################

eigenfaces\_matrix = matrix(summary(means\_pca)$sdev[1:9], 3,3, byrow=TRUE)

eigenfaces\_matrix

eigenfaces = pixmapGrey(eigenfaces\_matrix)

plot(eigenfaces)

#################

# Problem 2e

#################

summary(means\_pca)$loadings

means\_pca

means\_prin = princomp(means\_mat\_ref)

means\_prin$loadings[1,]

means\_prin$scores[,1]

dim(means\_pca)

yaleB05 P00A+010E+00.pgm

face\_05 = read.pnm(file = "CroppedYale/yaleB05/yaleB05\_P00A+010E+00.pgm")

plot(face\_05)

final\_matrix=NULL

p=1

for(i in 1:5){

face\_row=NULL

for(j in 1:5){

if(i==1 && j==1){

z=as.vector(means\_mat\_ref)

#z=rbind(z,z)

}

else {

means\_mat=means\_mat\_ref+((as.matrix(means\_pca$x[,p]))%\*%t(as.matrix(means\_pca$x[p,])))

mean\_face\_t=t(means\_mat)

p=p+1

face\_vec=as.vector(mean\_face\_t)

z=cbind(z,face\_vec)

}

}

final\_matrix=rbind(final,z)

}

faces = pixmapGrey(final\_matrix)

plot(faces)

as.matrix(means\_pca$x[,1])

t(as.matrix(means\_pca[1,]))

loadings(means\_pca)

head(means\_pca)

dim(means\_pca$x)

#################

# Problem 2e

#################

new\_matrix = faces\_matrix[5:152,]

dim(new\_matrix)

new\_pca = prcomp(new\_matrix)

summary(new\_pca)

final\_matrix2=NULL

p=1

for(i in 1:5){

face\_row=NULL

for(j in 1:5){

if(i==1 && j==1){

z=as.vector(means\_mat\_ref)

#z=rbind(z,z)

}

else {

means\_mat=new\_matrx+((as.matrix(new\_pca$x[,p]))%\*%t(as.matrix(new\_pca$x[p,])))

mean\_face\_t=t(means\_mat)

p=p+1

face\_vec=as.vector(mean\_face\_t)

z=cbind(z,face\_vec)

}

}

final\_matrix2=rbind(final,z)

}