STAT 5353: R Assignments 01

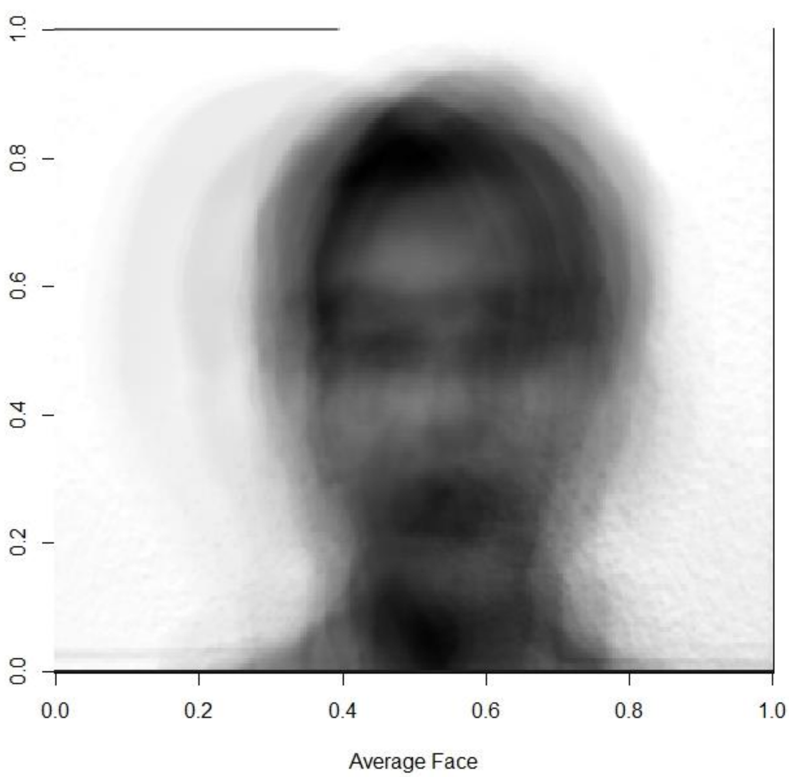
Upload your answers, as a single file, in Blackboard by 11:59 PM on Wednesday, Dec 15, 2021

PART-A

(a) What are the values of n and p?

Solution: The values of n and p are below:
n represents the number of observations, and we have 120 observations and p represents the number of covariates and X is a px1 random vector so, by using the yalefaces_train folder in Face_image data, in R we computed the values of n and p and got n =120 and p = 77760 many covariates.

(b) Plot the "average face".
Solution:

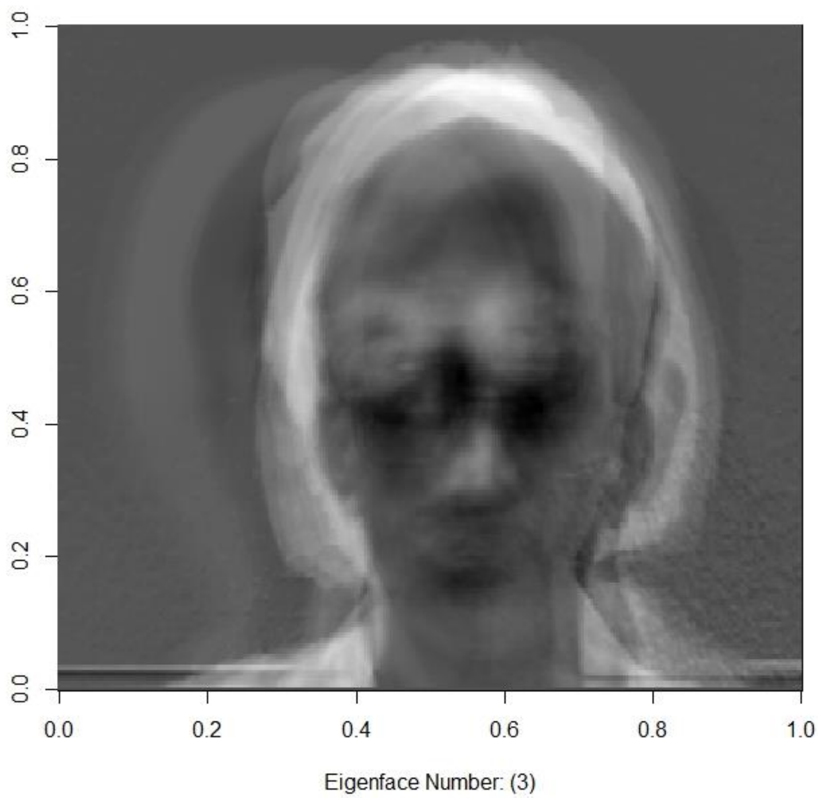


(c) Do a PCA allowing 25% loss of information. What is the value of k? Plot the third eigenface.
Solution:
We have to choose the minimum value of k that results in less than a prespecified 25% loss of variability, so we choose k = 8 and we need 8 principal components to perform PCA. The loss of variability formula is given by:

$$Lov = 1 - \frac{\hat{\lambda}_{(1)} + \hat{\lambda}_{(2)} + \cdots + \hat{\lambda}_{(k)}}{\hat{\lambda}_{(1)} + \hat{\lambda}_{(2)} + \cdots + \hat{\lambda}_{(p)}}$$

When k = 8 the lov is 0.238 which is less than given prespecified threshold 25%, so we choose k = 8.

The third eigen face is shown below:



(d) Now, choose any one image from yalefaces test folder (Every student's choice should be independent of any other student's choice: mention your chosen image name clearly in your written report) Reconstruct your chosen image using PCA from part (c). Plot the original image and reconstructed image

Solution: I have chosen the first image from the yalefaces test folder, and we perform the PCA with 8 principal components and reconstruct the approximate version of this test image.

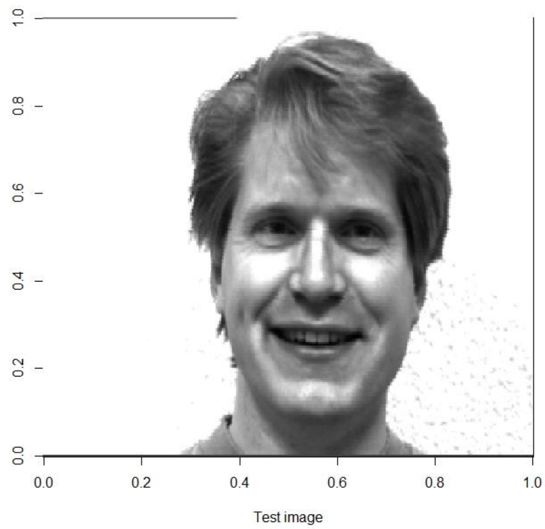
we can construct k-dimensional PC representation for an observation $X^{(new)}$ as:

$$Z^{(new)} = \hat{A}_k\left(X^{(new)} - \hat{\mu}\right)$$

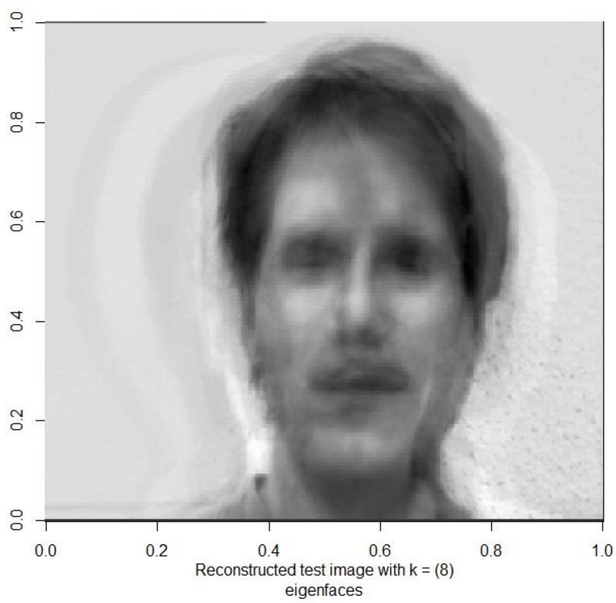
We construct the approximated version of the actual observation using the formula:

$$X_R^{(new)} = \hat{\mu} + \hat{A}_k Z^{(new)}$$

The original test image is shown below:



The reconstructed image of test image using PCA with 8 components is:



$\hat{\lambda}_1$

(e) Repeat (d) using the same image that you have chosen before but allowing 5% loss of information. What is the value of k now? Plot the reconstructed image. Compare between the qualities of two reconstructions that you get now with the one you got in (d).

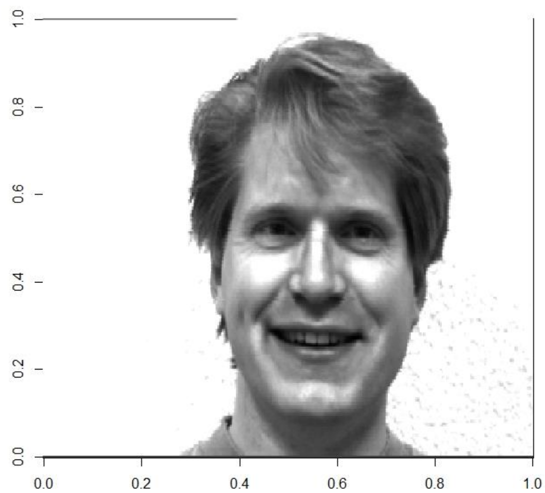
Solution:

We have to choose the minimum value of k that results in less than a prespecified 5% loss of variability, so we choose k = 38 and we need 38 principal components to perform PCA. The loss of variability is given by:

$$Lov = 1 - \frac{\hat{\lambda}_{(1)} + \hat{\lambda}_{(2)} + \cdots + \hat{\lambda}_{(k)}}{\hat{\lambda}_{(1)} + \hat{\lambda}_{(2)} + \cdots + \hat{\lambda}_{(p)}}$$

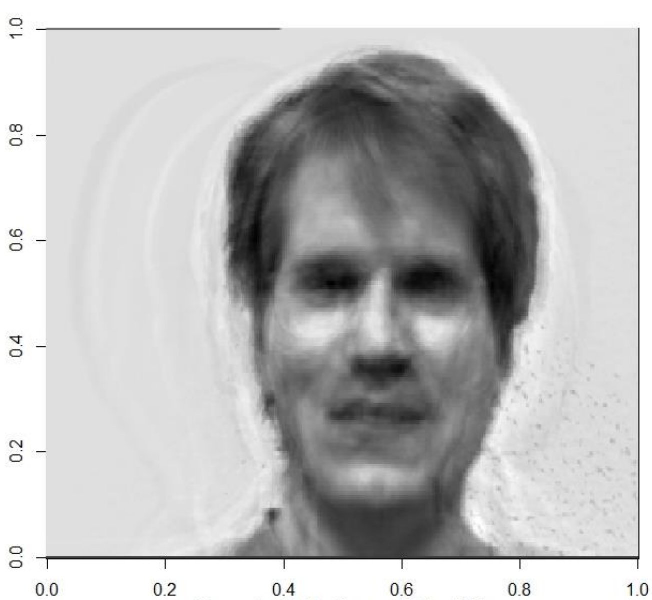
When k = 38 the lov is 0.0488 which is less than given prespecified threshold 5%, so we choose k = 38.

I have chosen the first image from the yalefaces test folder, and we perform the PCA with these 38 principal components and reconstruct the approximate version of this test image. The original test image is shown below:



Test image

The reconstructed image of test image using PCA with 38 components is:



Reconstructed test image with k = (38) eigenfaces

The quality we achieved by reconstructing the original test image with 38 principal components by performing Principal component analysis is much better than compared to the reconstructed version of this test image using 8 components in PCA we performed in part (d) because, as ‘k’ becomes larger, we expect the reconstructed image to be a better approximation to original image.

Part – B

```
## STAT 5353 class example for application of PCA to facial images
library(rgdal)
## Read all images from the folder
training_images_folder =
"C:/MS/FALL_2021/MULTIVARIATE/R_assignments/Faceimage_data/yalefaces_train/"
list_of_files = list.files(training_images_folder)

##################The value of n#################
n = length(list_of_files)
n
images_store = list()
for (i in 1:n)
images_store[[i]] = readGDAL(paste(training_images_folder,list_of_files[i],sep=""))$band1
image_size_rows = 243
x11()
imagedata = matrix(images_store[[1]],nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1, length=256)))
############## value of P######################
p = length(images_store[[1]])
p
x_data = matrix(0,n,p)
for (i in 1:n) x_data[i, ] = images_store[[i]]
## Compute the estimate of mu ## estimate the mean
x_bar = colMeans(x_data)
mu_hat = x_bar

################# plotting the avg face #############
## mu_hat is called the average face
x11()
imagedata = matrix(mu_hat,nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)),xlab="Average Face")
B = matrix(0,p,n)
for (i in 1:n)
{
 B[,i] = x_data[i, ] - x_bar
}
############## Finding eigenvectors of Sigma_hat ##########
## As n < p, use the following:
```

```
B_transpose_B = t(B)%*%B
eigen_results = eigen(B_transpose_B,only.values=F)
sum_eigs = sum(diag(B_transpose_B))
ELOV = array(0,n)
for (k in 1:n)
{
 ELOV[k] = 1 - sum(eigen_results$values[1:k])/sum_eigs
 print(paste(k," ",ELOV[k],sep=""))
}
## Choose k as the minimum integer for which ELOV < 25%
################# pca using k = 8 ########################
k = 8
eigen_kvalues = eigen_results$values[1:k]
eigen_kvectors = eigen_results$vectors[,1:k]
eigvec_Sigma_hat_k = matrix(0,p,k)
for (i in 1:k) eigvec_Sigma_hat_k[ ,i] =
c(1/sqrt(eigen_kvalues[i]))*(B%*%matrix(eigen_kvectors[,i],ncol=1))
Ak_hat = t(eigvec_Sigma_hat_k)
## if you want to plot any of the eigenfaces, plot corresponding row of Ak_hat
x11()
imagedata = matrix(Ak_hat[3, ],nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)), xlab="Eigenface Number:
(3)")
## read a new image X^(new) from a file
New_image =
readGDAL("C:/MS/FALL_2021/MULTIVARIATE/R_assignments/Faceimage_data/yalefaces_test/subjec
t01.happy")$band1
## plot this image file
x11()
imagedata = matrix(New_image,nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)), xlab= "Test image")
## Construct the PC representation of this image
PC_scores = Ak_hat%*%( matrix( New_image - mu_hat, ncol=1) )
## Reconstruct the original observations
Reconstructed_image = mu_hat + t(Ak_hat)%*%matrix(PC_scores,ncol=1)
## How good were the reconstructed observations: plot those reconstructed images
x11()
imagedata = matrix(Reconstructed_image,nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)), xlab="Reconstructed test
image with k = (8) eigenfaces")

## Choose k as the minimum integer for which ELOV < 5%
######################## pca using k = 38########################
```

```
k = 38
eigen_kvalues = eigen_results$values[1:k]
eigen_kvectors = eigen_results$vectors[,1:k]
eigvec_Sigma_hat_k = matrix(0,p,k)
for (i in 1:k) eigvec_Sigma_hat_k[ ,i] =
c(1/sqrt(eigen_kvalues[i]))*(B%*%matrix(eigen_kvectors[,i],ncol=1))
Ak_hat = t(eigvec_Sigma_hat_k)
## if you want to plot any of the eigenfaces, plot corresponding row of Ak_hat
x11()
imagedata = matrix(Ak_hat[3, ],nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)), xlab="Eigenface Number:
(3)")
## read a new image X^(new) from a file
New_image =
readGDAL("C:/MS/FALL_2021/MULTIVARIATE/R_assignments/Faceimage_data/yalefaces_test/subjec
t01.happy")$band1
## plot this image file
x11()
imagedata = matrix(New_image,nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)), xlab= "Test image")
## Construct the PC representation of this image
PC_scores = Ak_hat%*%( matrix( New_image - mu_hat, ncol=1) )
## Reconstruct the original observations
Reconstructed_image = mu_hat + t(Ak_hat)%*%matrix(PC_scores,ncol=1)
# How good were the reconstructed observations: plot those reconstructed images
x11()
imagedata = matrix(Reconstructed_image,nrow= image_size_rows,byrow=T)
image(t(imagedata[nrow(imagedata):1,]),col = grey(seq(0,1,length=256)), xlab="Reconstructed test
image with k = (38) eigenfaces")
```