Anamol Acharya

R11651561

Data Analysis

Assignment 2

**Question 1:**

Eigen Faces:

**Key idea:**

Eigen face is a method of creating an imaginary image which can be determined by adding the mean face from the pool of images. Eigen faces can be used to reconstruct/identify the faces by determining the variance of the faces in the dataset of the larger number of images. Normally, the initially phase of determining the eigen face, the raw image (say: human) might look like the ghost or the x-ray report of the face. PCA is used to calculate eigen faces by estimating the principal components of the image dataset.

We assume that the most face images lie on a low dimensional sub spaces in the big image space determined by the first k eigen vectors (k<<< d) direction of maximum variance. So, it uses PCA to find the vectors/eigen faces u1,u2, … uk that span that subspace. It represents all face images in the dataset as linear combinations of eigenfaces. Find the coefficients by dot product.

**PCA and Eigenfaces**

Data reduction occurs among multiple number of images using PCA. Coefficient of all eigen vectors is obtained. Represent the entire image by multiplying Coefficient of the eigen vectors \* eigen vector will give the new image. While performing the projection of the eigen faces, mean image not clear since it is an average of all the images from the pool. Mean is subtracted out from the distribution. Then, start computing the eigen vectors. Using PCA, the dimension of the images are reduced in a feasible way obtaining the eigen vector, which later can be used to obtain the eigen faces.

**The steps used to undertake eigen faces based face classification is given below:**

Input: dataset of N face images

Face: k\* k bitmap of pixels

To do anything we have to form a single vector out of the dataset of the images.

Unfold the unfold the first row of the pixels, they will be first k attributes.

We end up a vector with the k^2 row of dimensions.

We arrange in a matrix () k^2\*N

Then PCA is done, resulting eigen vectors.

Set of m eigenvector(columns) each is K^2 dimensional.

Take one of the eigen vectors, then fold it back into a K\*K bitmap, so we take first k attribute, make it into the row of pixels. – The image loos scary, with the similarity of the human faces (variation- dimension to the people who vary the most).

**Eigen Faces: Projection**

Project new face to space of eigen-faces

Represent vector as a linear combination of principle components.

So, to get the real picture = mean + sum of (some component \* some amount of 1 eigen face)

The formula can be mathematically represented as:

**F(final) = F(mean) + ;**

F(final) is a final new face,

F(mean) is the mean of the average face of the pool of bitmap image,

F is the eigen face of each multipliers(components),

α is the scalar multipliers of each components

Component can be positive or negative.

As we increase number of eigen vectors gradually, we will more likely see the image improving respectively looking closely to the original image.

We can use to massive application, if we all have same eigen vectors, all we need is Projection coordinates, then we can transcript the arbitrary image.

This give the huge reduction of the data and decrease the redundancy.

Application: Can be used in face similarities, finding facial ancestries and many more.

Cons: The new dimensions form the average of the eigen vectors, a rare image of different animal or object looks like the human face.

**Report for Question 2**

Outline of the code.

1. Read the public dataset of images that was imported from the GitHub.
2. Images are visualized using the dataset. The images are displayed, but the image are in tilted form. So, the images are rotated 90 degree to make straight up.
3. For loop is used to straight up all the images in the dataset.
4. The rotated images from the dataset are saved in a different csv file that in save in the given directory.
5. The mean value of the images is calculated. This means the certain part of the images is common in all images is calculated.
6. With the average face calculated, it is subtracted from all images data provided.
7. The covariance matrix is calculated and PCA is performed
8. After, determining the covariance matrix of the images, eigenvalues and eigen vectors are calculated. The largest 40 eigen value and its corresponding eigen vectors are determined.
9. Now, the eigen faces projection is performed where, 1st, 11th, photo are projected into the eigen spaces. Here, the large set of pool of dataset is reduced to small set of data using PCA. Each and every image has different projections in eigenvector space.
10. The formula to obtain the eigen faces is implemented in the following code.

The formula can be mathematically represented as:

**F(final) = F(mean) + ;**

F(final) is a final new face,

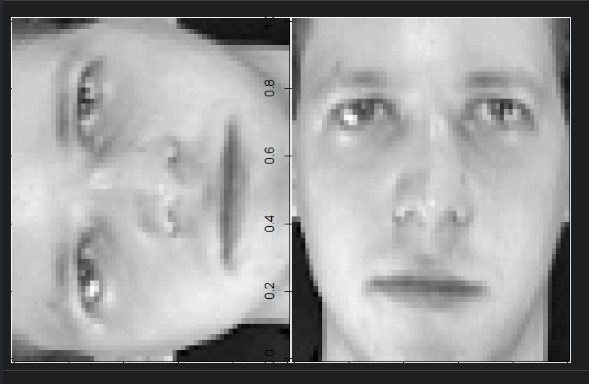
F(mean) is the mean of the average face of the pool of bitmap image,

F is the eigen face of each multipliers(components),

α is the scalar multipliers of each components

1. The image of the person is reconstructed following the formula of eigen faces. With multiple images constructed, the average face is also calculated in the display.

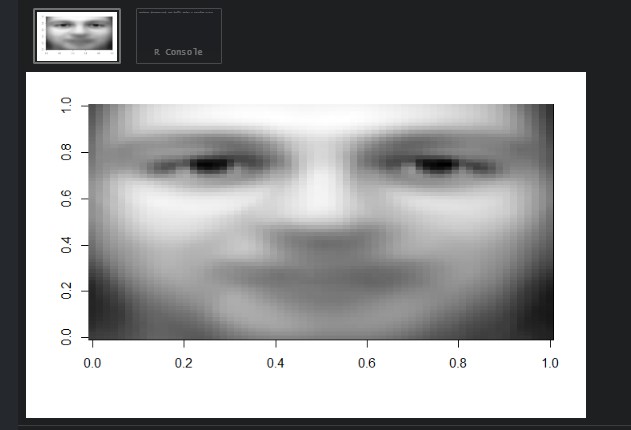
1.



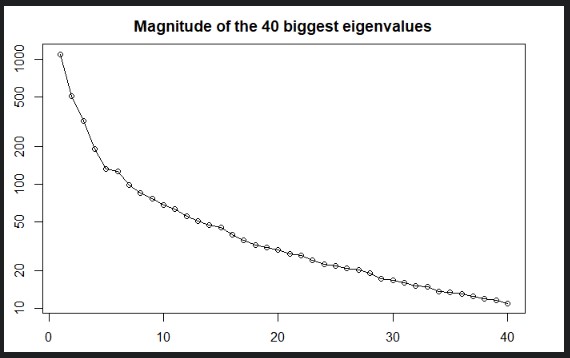
2.



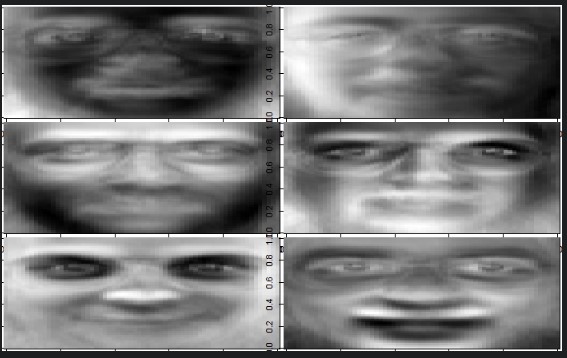
3.



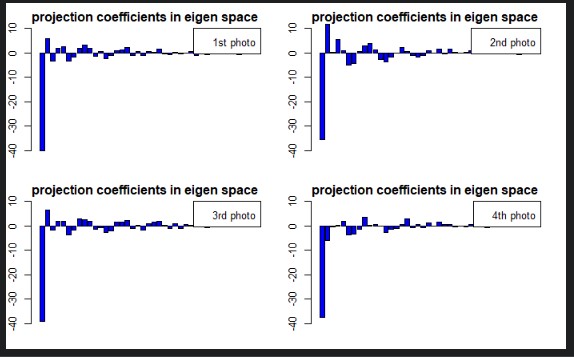
4.



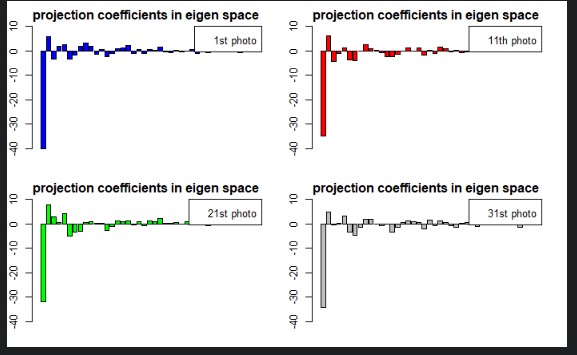
5.



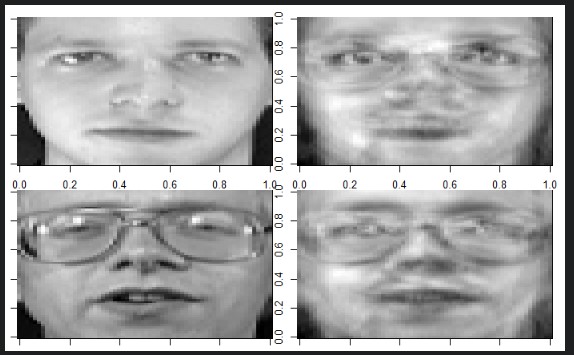
6.



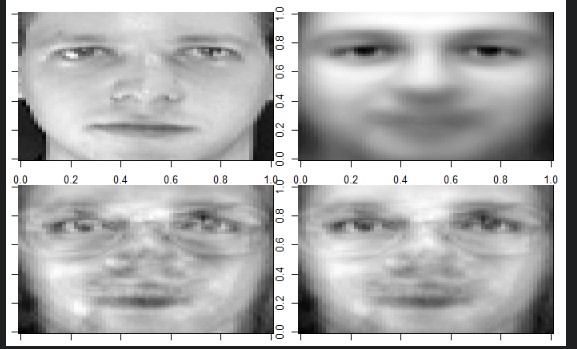
7.



8.



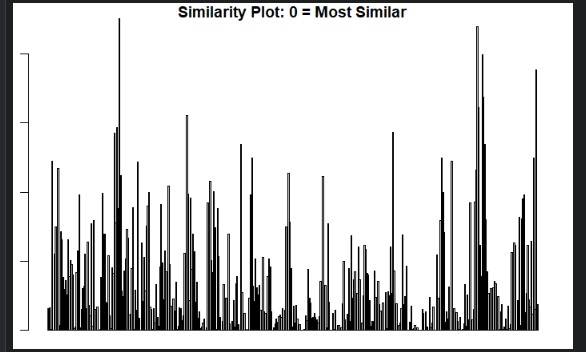
9.



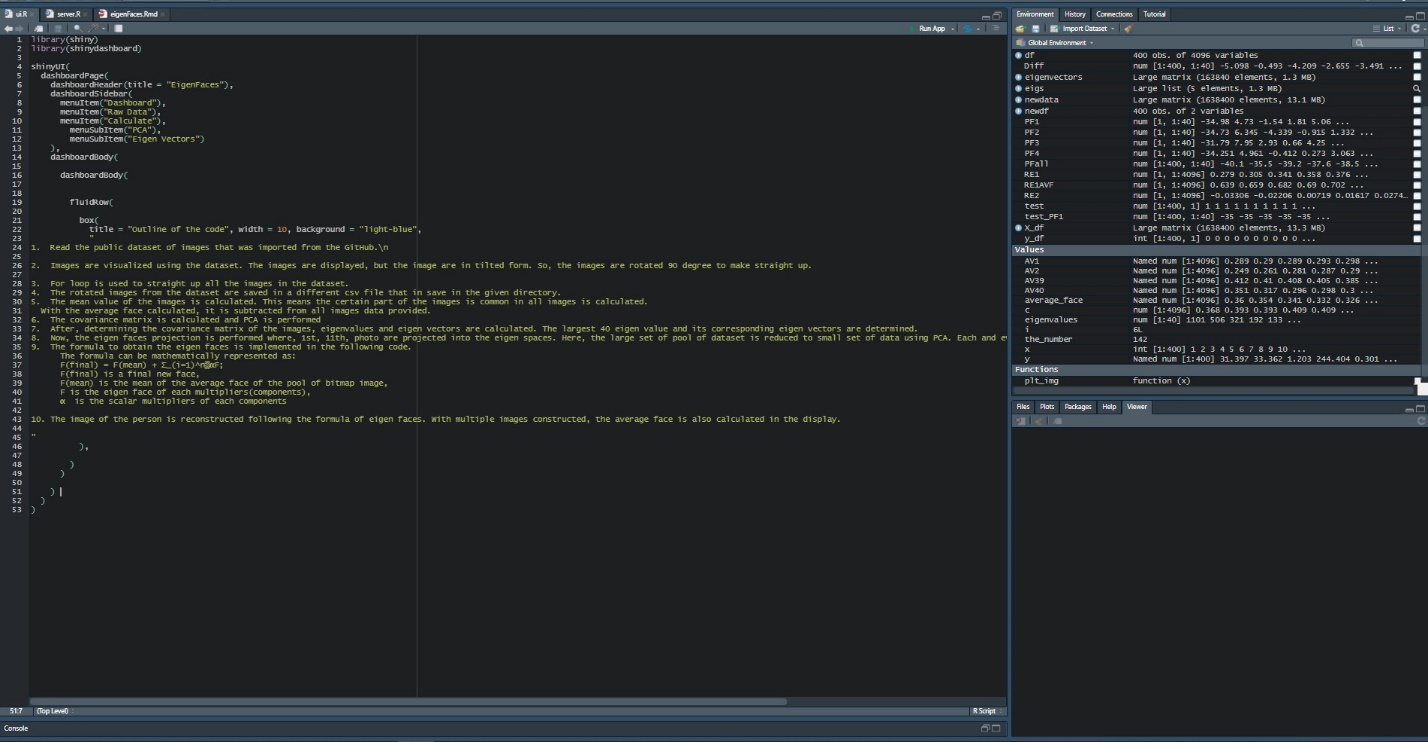
10.



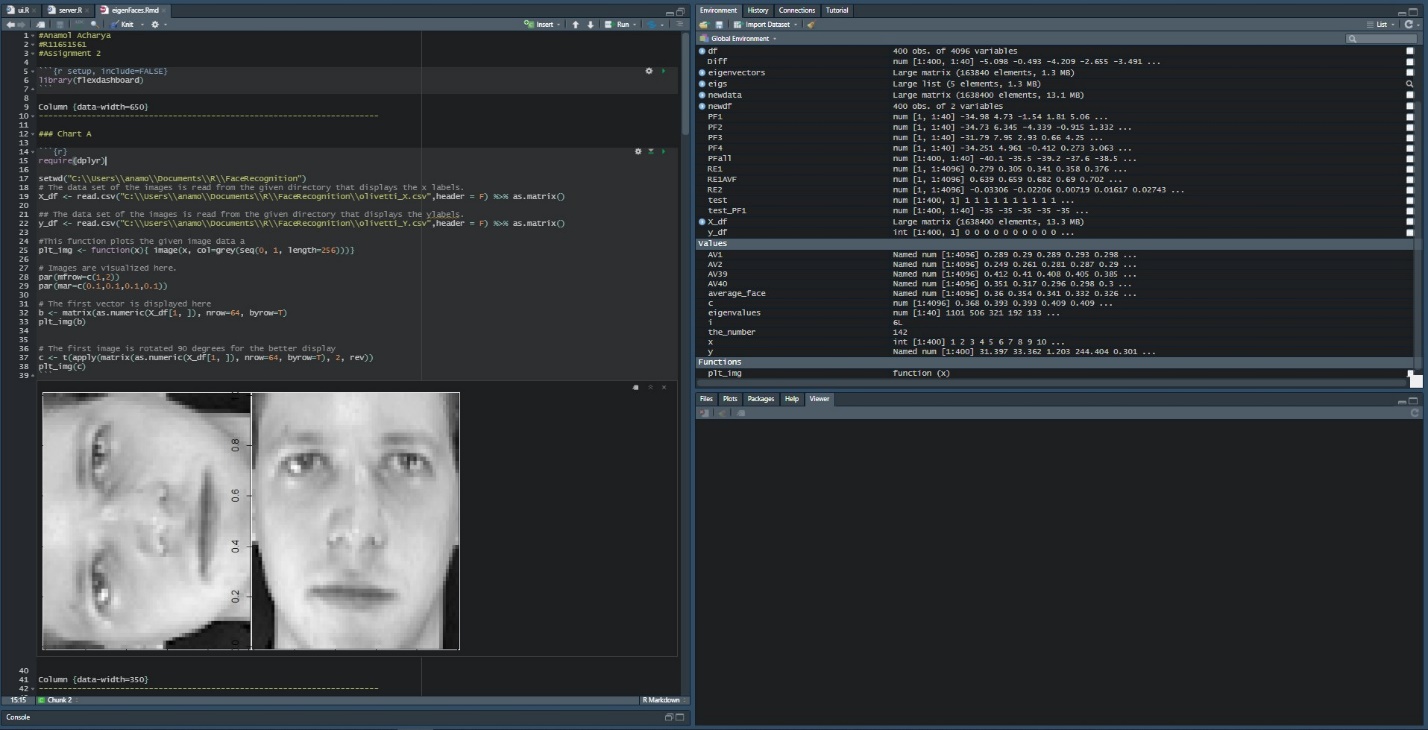
11.



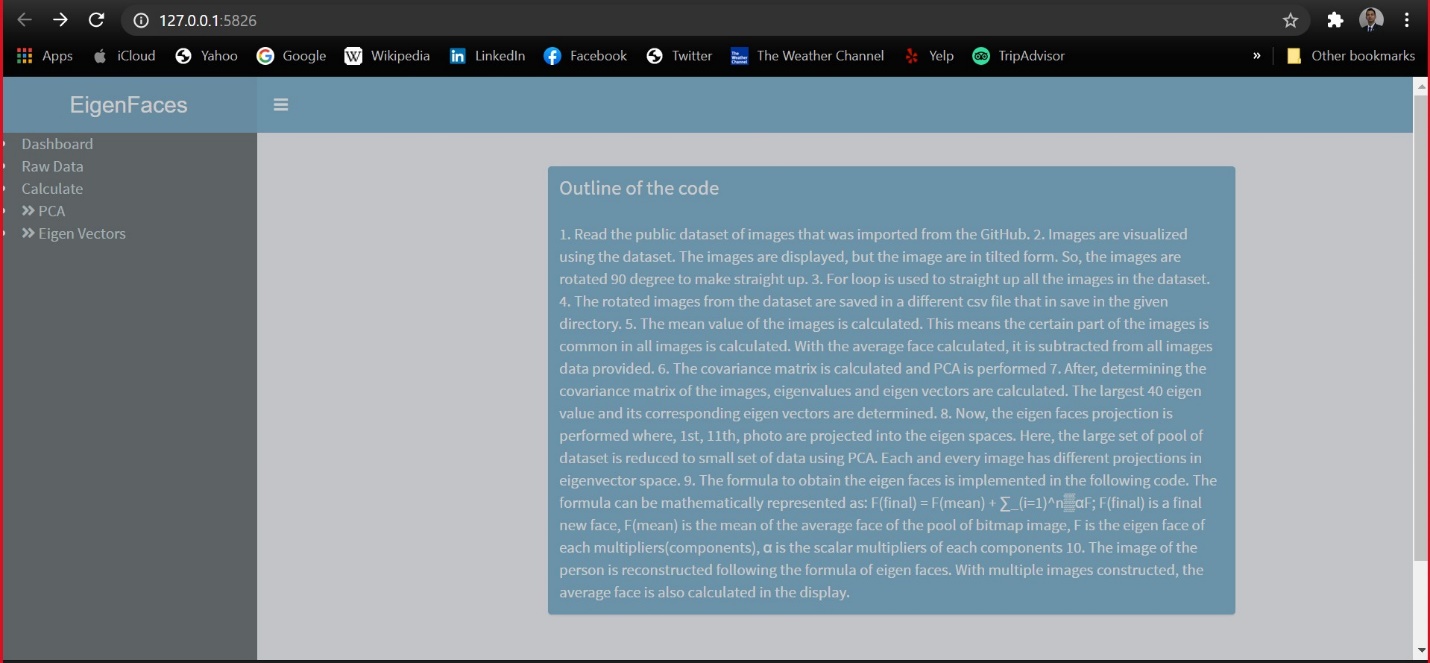
12.



13.



14. UI layout



The R studio need to import all necessary packages like Shiny, Shiny dashboard.

Then, the directories need to be corrected from the user’s PC to make sure the data set is read properly.

The Rmd file need be run where chunk of R codes are run turn by turn to see the result of eigen faces and PCA.

The research was conducted by referring following link that contributed the dataset and algorithm concepts.

Thank you very much.

Reference:

https://rstudio-pubs-static.s3.amazonaws.com/469844\_332494e669e1439a8ab3f8ea489459c4.html

https://rstudio.github.io/shinydashboard/

Dataset reference:

https://github.com/rajsiddarth/Datasets