Homework 1

Statistics W4240: Data Mining
Columbia University
Due Friday, February 5 (Section 01)

Problem 1. (50 Points) We are going to load the data from the Yale Faces B data set, which is in the Files and Resources tab of Courseworks. These are .pgm images, which are viewable either through R or with a specialized viewer like Gimp. There are 38 subjects (labeled 1 to 39 with no 14), each photographed in a variety of lighting conditions. The file name denotes the data set, the subject, and the lighting condition. We will look at three lighting conditions, `P00A-005E+10`, `P00A-005E-10`, and `P00A-010E+00`, which are closest to straight on lighting. We will use the `pixmap` library to manipulate the data. Load this library and make sure that the folder `YaleCropped` is in your working directory. You should begin by downloading `hw01_partial.R` from Courseworks or Piazza, which will give you a template. You will then need to fill out key sections of this code; each of these sections is delineated by the comments `#-----START YOUR CODE BLOCK HERE----#` and `#-----END YOUR CODE BLOCK HERE----#`.

a. (10 Points) Load the picture `yaleB01_P00A-005E+10.pgm` with the command:

```
face_01 =read.pnm(file = "CroppedYale/yaleB01/yaleB01_P00A-005E+10.pgm")
```

You can view the image with the command

```
plot(face_01)
```

What class is `face_01`? What is the size of the original image in pixels?

b. (10 Points) Make `face_01` into a matrix with the command:

```
face_01_matrix=getChannels(face_01)
```

Using the same steps above, you can load and create a second image matrix `face_02_matrix`. You can then concatenate images in the following way:

```
faces_matrix=pixmapGrey(data=cbind(face_01_matrix,face_02_matrix))
```

What is the maximum value that a pixel can take for this type of file? The minimum value? What colors do those values correspond to?

c. (10 Points) Let's load in all of the data by looping through the folders and storing the values in a list. Before we start that, run the following commands:

```
dir_list_1 = dir(path="CroppedYale/",all.files=FALSE)
dir_list_2 = dir(path="CroppedYale/",all.files=FALSE,recursive=TRUE)
```

What is contained in each of these variables? Give the number of elements and some example elements.

d. (20 Points) Read the data into a list (or set of lists) by looping through the folders. Instantiate the list before you read in the data. You can do this by concatenating strings (although there are other methods as well). String concatenation can be done with the following code:

```
pic_list = c( 09 , 12 , 22 )
view_list = c(  ’P00A-005E+10’ , ’P00A-005E-10’ , ’P00A-010E+00’)
i = 1
j = 3
filename = sprintf("CroppedYale/%s/%s_%s.pgm",
     dir_list_1[pic_list[i]] , dir_list_1[pic_list[i]] , view_list[j])
```

This will produce the string `filename` with the value `"CroppedYale/yaleB05/yaleB05_P00A-010E+00.pgm"`. After you have read in the .pgm files for views `P00A-005E+10`, `P00A-005E-10`, and `P00A-010E+00`, convert each of these to a matrix. Using the matrices in a loop structure, make an array of the pictures, where each row has one subject and each column has one view. Use subjects 09, 12, and 23 for the rows, and views `P00A-005E+10` in the left column, `P00A-005E-10` in the center, and `P00A-010E+00` in the right column. This produces a 3-by-3 grid of photos. Save the result as a .pdf and include it in your write up.

Problem 2. (50 Points) Let’s do a bit more data manipulation and learn how to use external resources to write code. There will be some functions that were not discussed in class. Use resources like `R` documentation and StackExchange to learn how to use these functions.

a. (10 Points) Load the views `P00A+000E+00`, `P00A+005E+10`, `P00A+005E-10`, and `P00A+010E+00` for all subjects. Convert each photo to a *vector*; store the collection as a matrix where each row is a photo. What is the size of this matrix?

b. (10 Points) Some of the items in `dir_list_2` are not image files and some of the images are corrupted. Corrupted images have a .pgm.bad extension. Generate a list of all valid image names. What is the length of this list? What is the 10th element? (Hint: use the commands from Q1c to generate a list of all photos, and the function `grepl()` to test if a file name contains .pgm but not .bad)

c. (20 Points) The flash failed to go off in a few of the photos, making the images almost all black. Use a threshold of 0.05 as a maximum pixel value. Give a list of the image names where this occurred, and give the range of pixel values for these photos (e.g. minimum and maximum values).

d. (10 Points) Vary the “no flash” threshold from 0 to 1. Make a plot with the threshold value on the x-axis and the proportion of “no flash” photos on the y-axis. What do you think is a good threshold? Why? (Hint: you might want to view the selected photos and file names for each threshold value.)