Homework 1

Statistics W4240: Data Mining
Columbia University
Due Tuesday, September 16 (Sections 01, 03, 04)
Due Monday, September 15 (Section 02)

For material from the James book, print out what appears on your R screen if there are no values requested (such as 2.8 (a)). For your .R submission, place each question in a separate .R file labeled `hw01_q1.R`, `hw01_q2.R`, and so on.

Submit all homework material (both .pdf and .R files) through Courseworks. It is due by 11:55PM on the due date listed for your section. Once you have uploaded your files, check them for errors.

Problem 1. (20 Points) James 2.8

Problem 2. (20 Points) James 2.9

Problem 3. (20 Points) James 2.10

Problem 4. (40 Points) We are going to load the data from the Yale Faces B data set, which is in the Resources tab of the Piazza Course Page. 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_q4_partial.R` from 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. (6 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. (7 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. (7 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( 05 , 11 , 31 )
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, make an array of the pictures, where each row has one subject and each column has one view. Use subjects 05, 11, and 32 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.