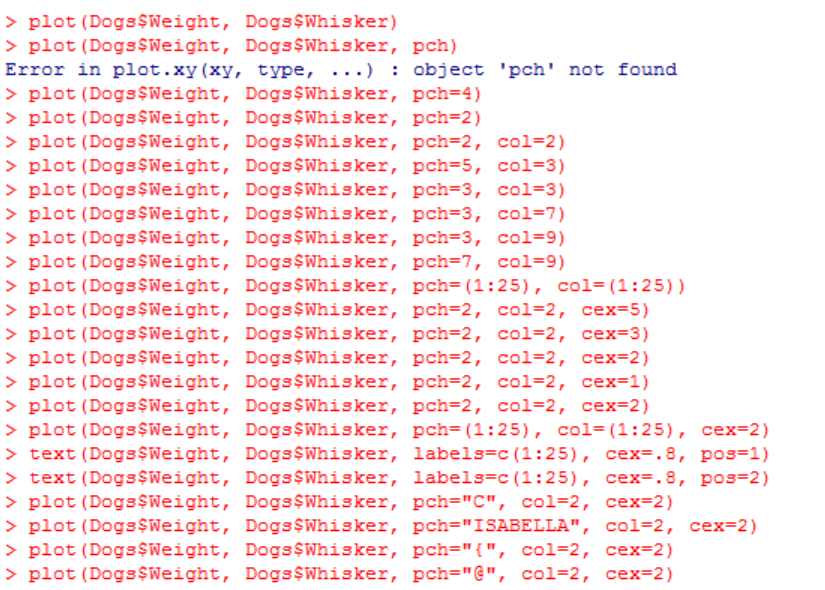
Yucehan Kucukmotor ***(Submitting late due to illness. We had e-mailed about this)***

CIS3920, Data Mining

Prof. Lawrence Tatum

Due 10/07-~~Submitted 10/14(canceled)~~, submitted 10/21

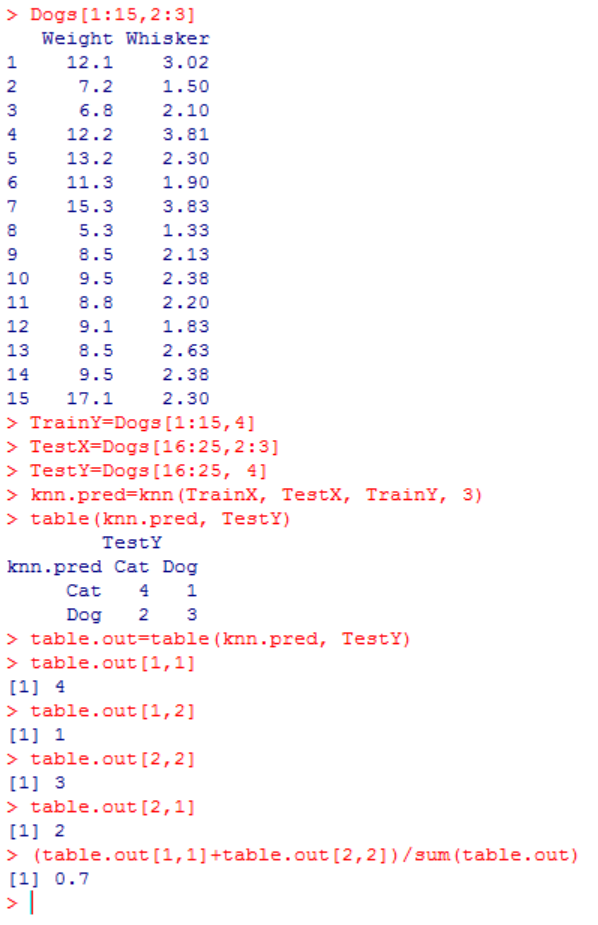
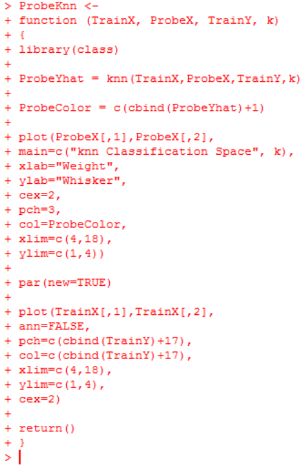
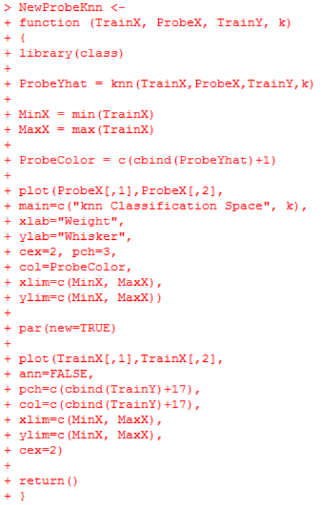
Prework:

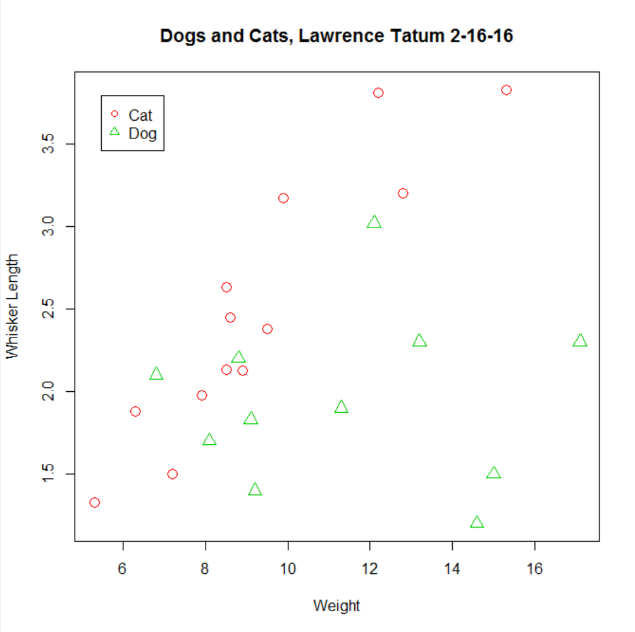
I started off by getting familiar with the work and the way a graph is styled by trying pch, col, cex with different values as well as the labels function. This helps me internalize what does what and helps me save time in the future.

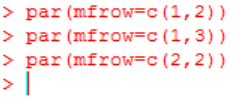
On the bottom left, I have a cropped screenshot which is more relevant to the previous lecture. However, I had asked Professor Tatum after class in the computer lab of the 6th floor how to define the

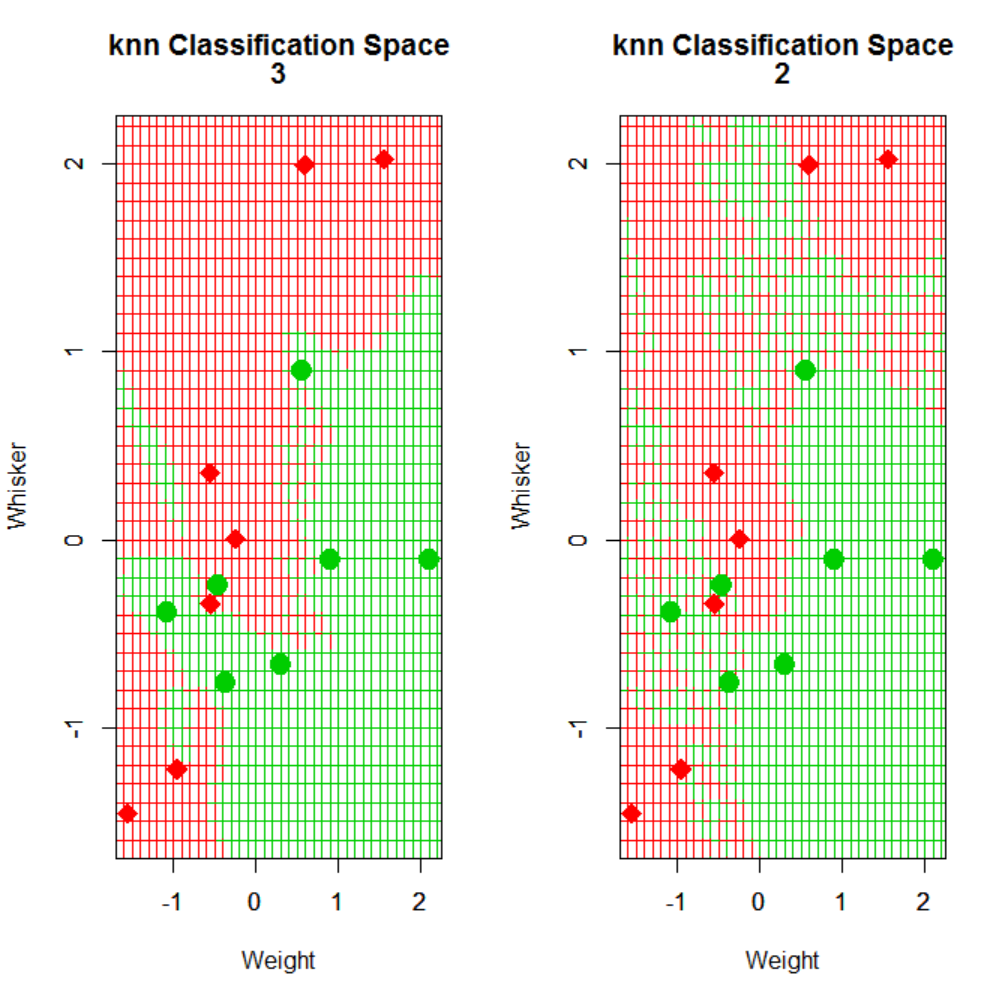
Knn function properly since I had failed to define it and would get errors every time I tried to have my knn.pred defined. I got it worked with the help of Professor Tatum, and redid my work so that I would have objects and functions ready for the lectures and homework to come. Afterwards, as can be seen on the right, I have copied the code from Professor’s work as it was (not literally copied with ctrl+c, but typed it so that I’d familiarize myself with it) to see how a legend is created.

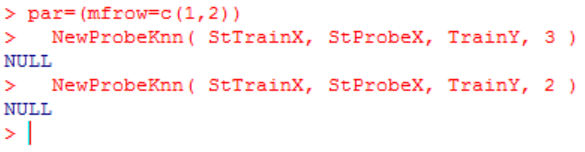
When creating the functions,

I did not actually type down

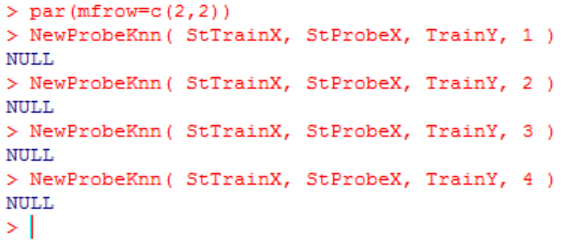
the code myself, but copied from Professor Tatum’s lecture notes, then I read them with my eyes. The reason of copying the functions was to ensure they would be same as the Professor’s and that I would not encounter unexpected errors due to a minor problem I might have caused otherwise.

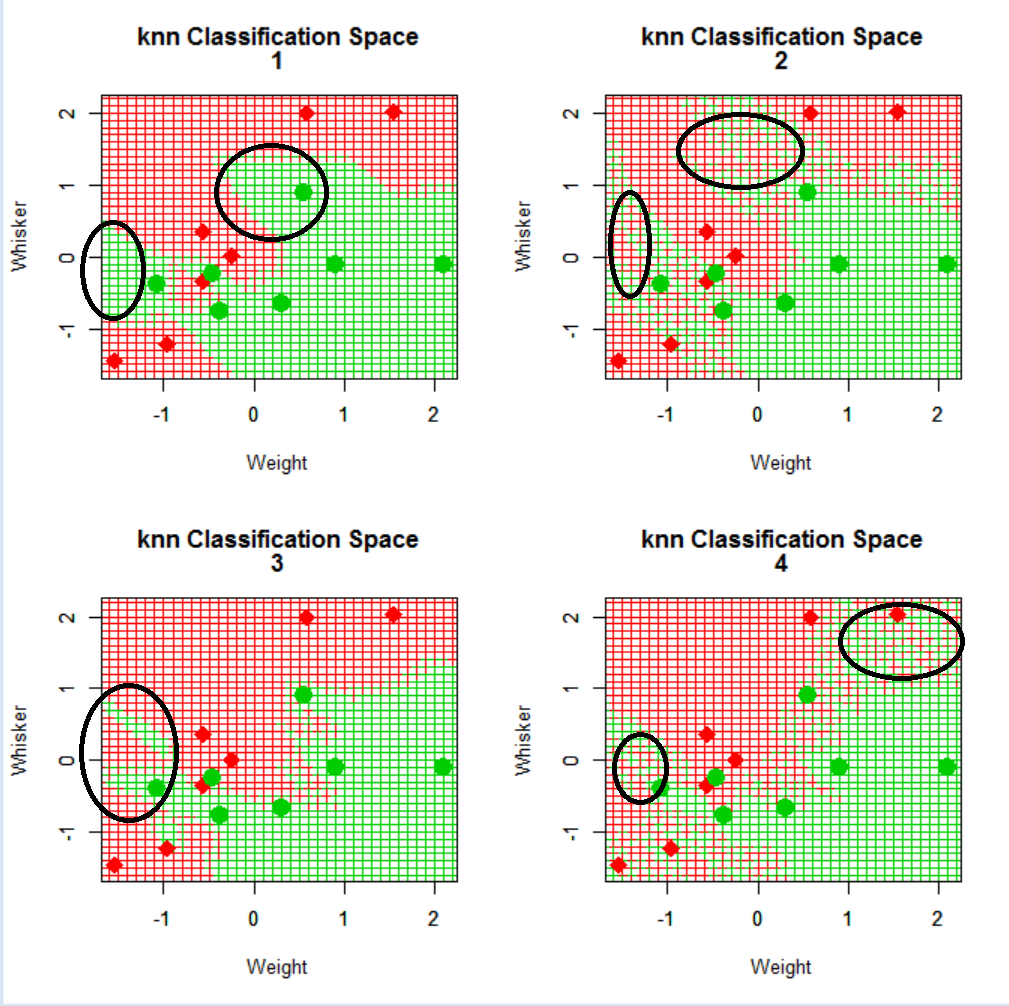
5.1 ) I started off by exploring what par(mfrow) is to know what I am doing. Turns out par(mfrow) function is very useful and handy for creating a simple multi-paneled plot (layout function can be used for customized panel plots of varying sizes). On the right, I did exercise on par function to see the results for different number of rows and columns for plotting.

On the right I have two plots in one figure. One for k=3, the other for k=2.



To get a 2 by 2 formation, we need c(2,2) and then plot what we want to plot one by one as seen below. par(mccol) can be used as

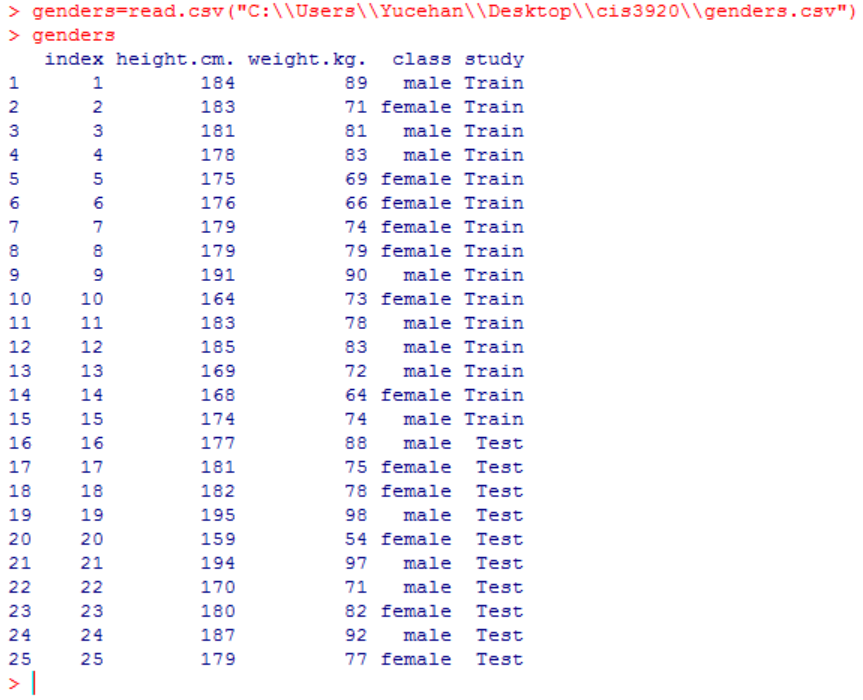
well if we want to place our graphs by filling the column first and then moving on to the next column (we would still have the same plots, different locations within the figure).



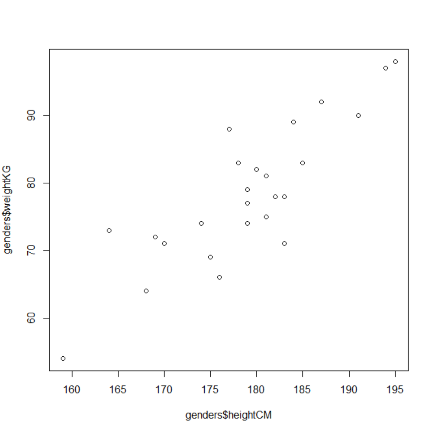
On the right is the 4 plots inside one single figure. We see that from k=1 to k=4 we get different amount of red and green in the background, scattered in a different fashion depending on the size of the k value we are using. K=2 and k=4 somewhat look similar when it comes to how the green in the background is spread and scattered. It is interesting how there is more scattered patchy greens when k=2 and k=4 as opposed to when we have our k value as 1 and 2.

To get back to 1 by 1 formation (to reset the grid), we can simply type:

par(mfrow=c(1,1))



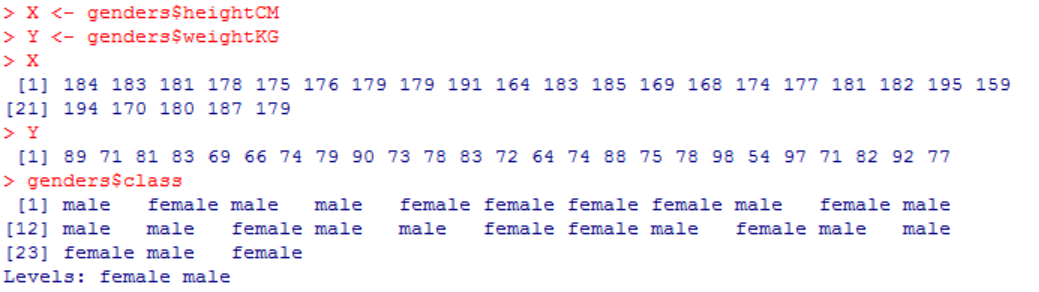
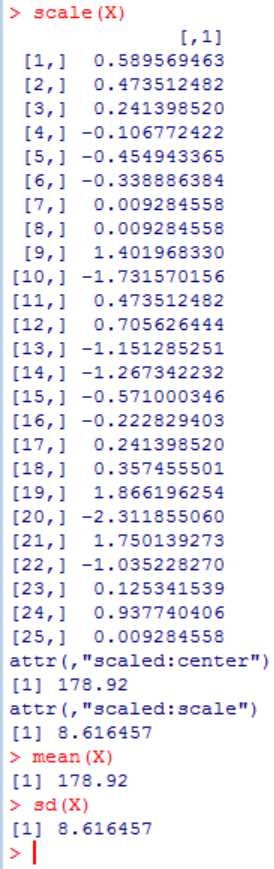
5.2) I created a .csv file containing 25 humans. Class is either male or female. Heights and weights are in centimeters and kilograms, respectively.

# Using plot parameters in R (using data set from genders.csv)

plot(genders$heightCM, genders$weightKG)

it seems like genders.csv file was successfully read into R.

# to standardize a column, we need to call scale function

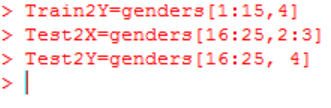


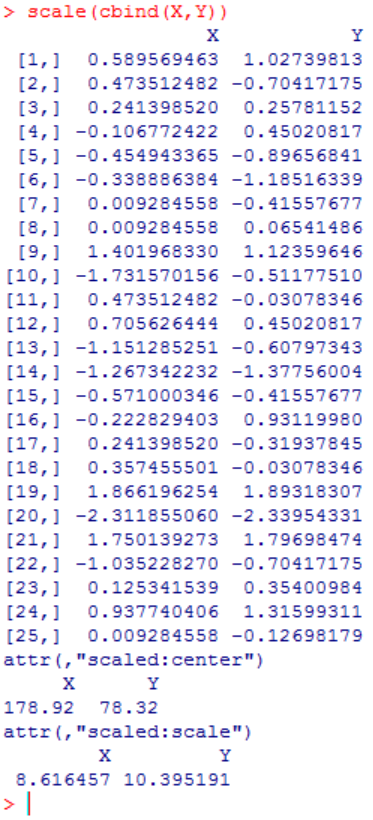
# X is easier to use than genders$heightCM an

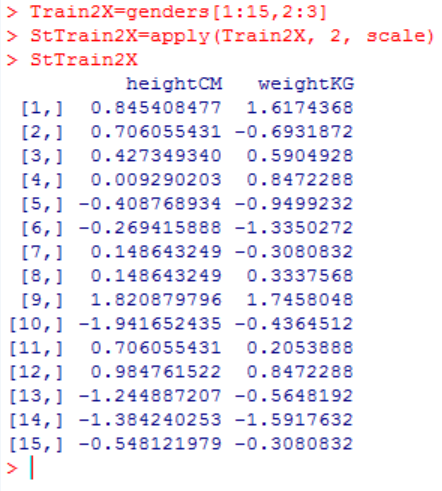
On the right I started off by binding X and Y with my values and then checked my values by typing X and Y respectively to see if I was getting the correct values.

I did call scale function to see standardized values of X and Y, and then only X. Numbers at the bottom are the mean and standard deviation respectively. Then I typed:

mean(X) and sd(X) in R to see if the mean and standard deviation I was given by the function scale were correct.

#assigned names to proper rows and columns.





I assigned rows 1 to 15 and columns 2 to 3 to Train2X (not to mistake it with TrainX).

#assign class values to a vector #green triangle=male

malefemale=genders$class #red circle=female

> plot(main="X is height in CMs and Y is weight in KGs",X,Y, pch=c(malefemale), col=(c(malefemale)+1) , cex=1.5)

Used above line to plot my graph. Recall that I had assigned my genders$height and genders$weight columns to X and Y to make future work easier.

# add legend to the graph to make it more readable to the viewer

> plot(main="X is height in CMs and Y is weight in KGs",xlab="Kilograms",ylab="Centimeters",X,Y, pch=c(malefemale), col=(c(malefemale)+1) , cex=1.5)

> legend("topleft", inset=.05, legend=c("female", "male"), pch=c(1,2), col=c(2,3))

#plot with the legend on topleft

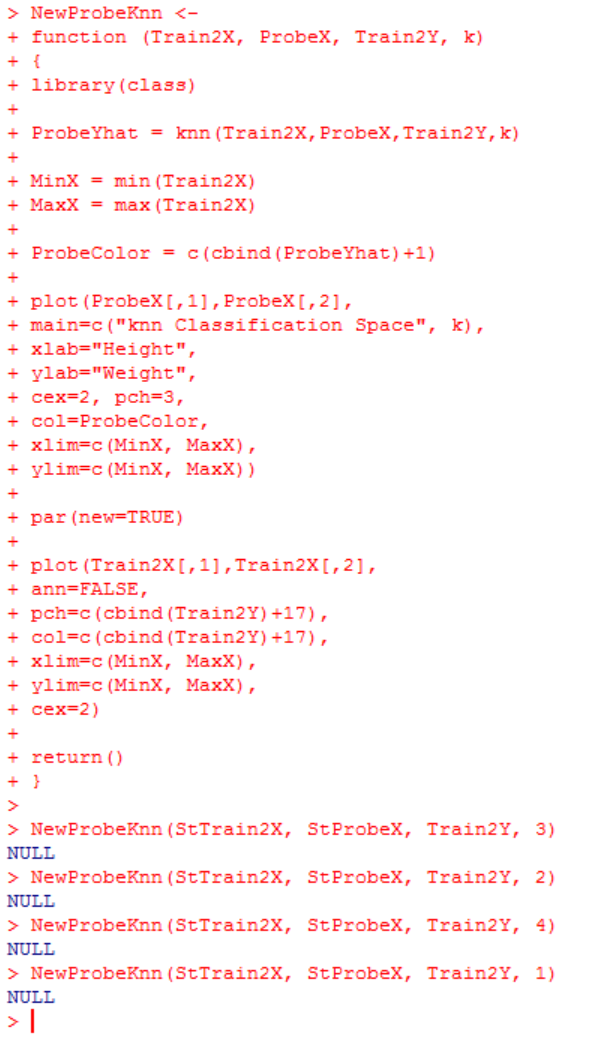
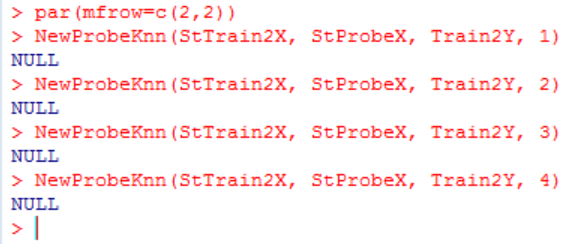


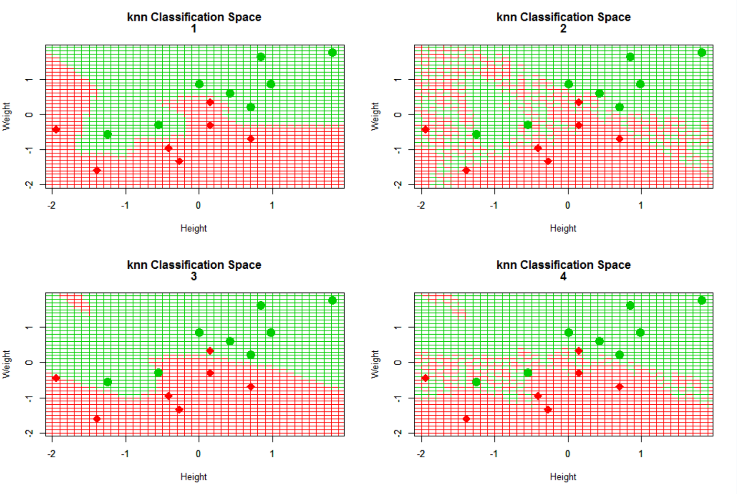
~~Note: I have not been able to run probe function because I am not sure as to how can I adapt ProbeX into my own work.~~

~~There are two columns in ProbeX. WeightTestX and WhiskerTestX. If I am to have ProbeX for my ‘genders’ work, then I would need two columns named HeightTestX and WeightTestX respectively. Yet, I am not sure if numbers I choose matter or not. It would not make sense to have heights going as 150.1,150.2,150.3,…,189.8,189.9,190 and so on since they are not actually used as measurement since centimeter is small enough measure on its own, 0.1 of a centimeter would be so insignificant. This is where I got lost and failed to run probe function. NewProbeKnn function needs 4 parameters. I have defined Train2X, Train2Y, and k is…k. Without ProbeX, I am not able to run the function.~~

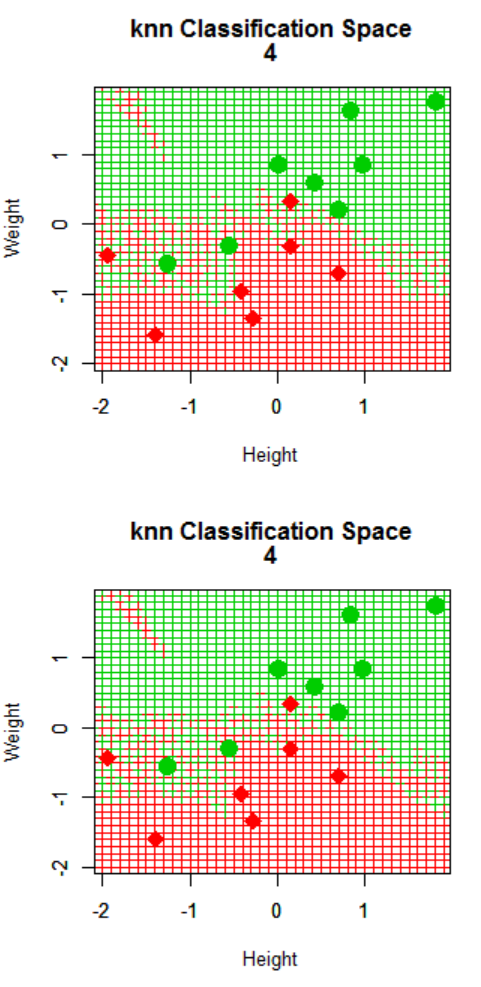
### After a break of 1 week, I am redoing this part.

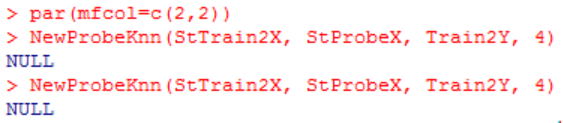
I’ve edited the NewProbeKnn function, changing the variable names from TrainX, TestX to Train2X and Test2X and so on to distinguish them from one another. Any other time when I need to run the function, I’ll simply refer to the lecture notes or just change the names back to what they were (TrainX, TrainY, TestX).

To the left is the lines of codes to show how I modified the NewProbeKnn function to have it function for my data sets and variable names (Train2X, Train2Y…).

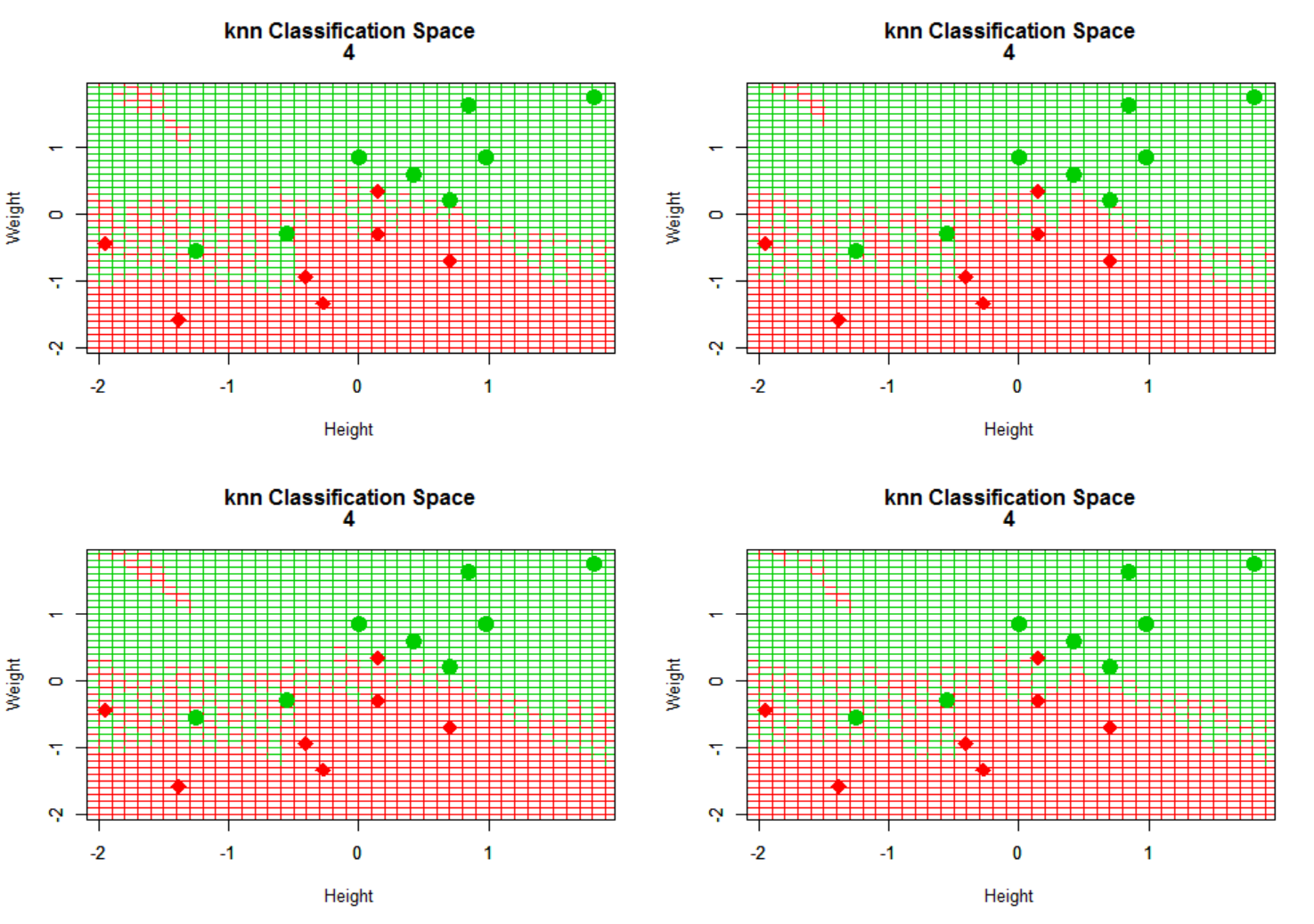


To the right is the lines of codes to show how I put the 4 graphs into one single frame using par function. (Note that mfcol also can be used instead of mfrow which would merely change the way graphs are deployed onto the frame, but nothing else. Especially content wise).





5.3) Note that I have used mfcol instead of mfrow to fill the column first instead of filling up the row first. Result can be seen on the right.



(k=4)

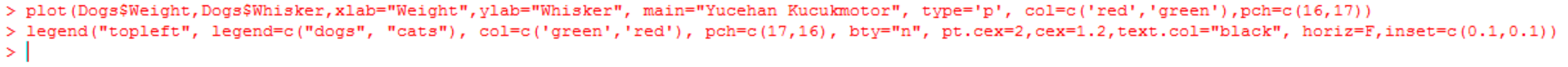
They seem to be almost identical. But I believe key word here is almost. As much as they seem like twin brothers and sisters, they do seem to have

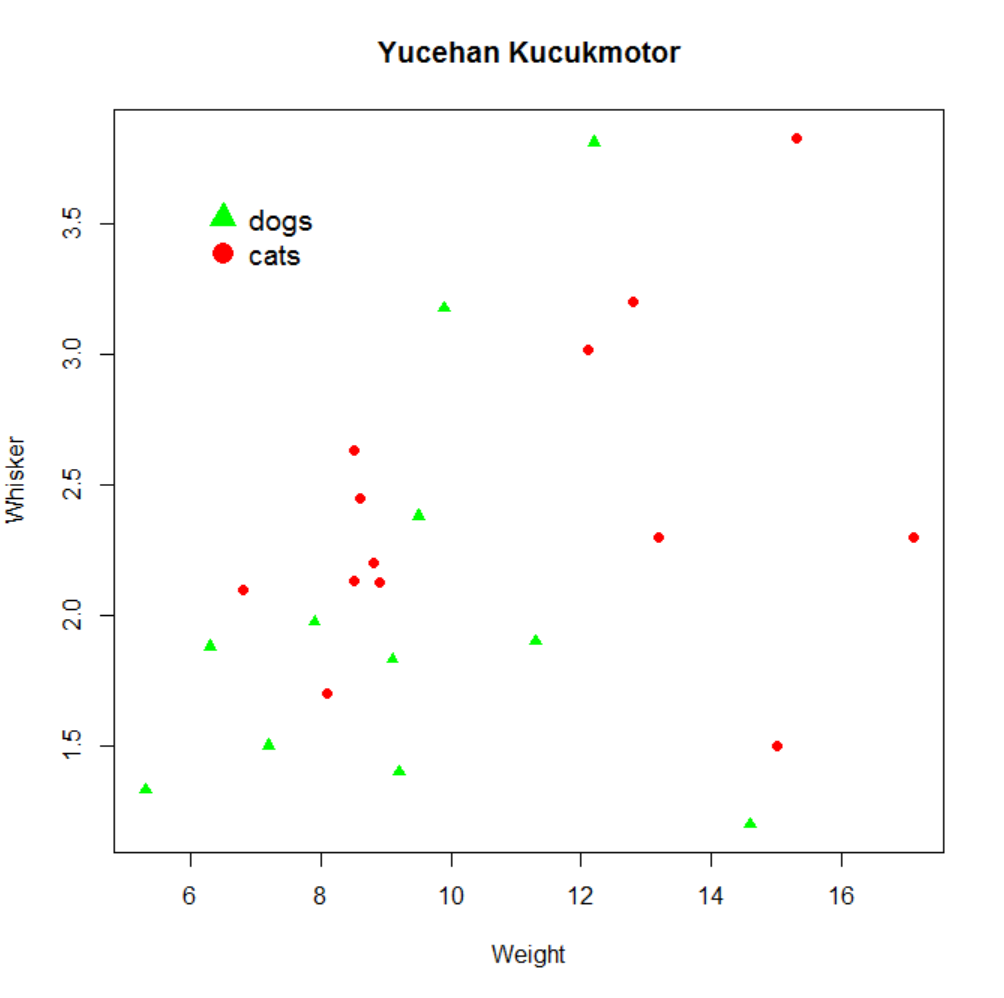
Very minor differences

On their own with very

Slight differences on how

Background color is scattered or spread on each graph.



5.4)