Week 2 – First Breakout: Sorting Data and Ordering a Data Frame

Instructions: In your group, introduce yourselves to each other again and then solve each of the problems below: I may ask you the names of the people in your group! All students in the group should be prepared to report the whole class at the conclusion of the breakout group. All students should run the necessary code on their own instances of R-Studio.

1. Make a copy of the built-in iris data set like this:  
   *myIris <- iris*
2. Get an explanation of the contents of the data set with the help function:  
   *help("iris")*
3. Summarize the variables in your copy of the data set, like this:  
   *summary(myIris)*
4. The summary() command provided the mean of each numeric variable. **Choose the variable with the highest mean** and list its contents to the console. Any variable can be echoed to the console simply by typing its name. Here’s an example that echoes the variable with the lowest mean to the console:  
   *myIris$Petal.Width*
5. Now repeat the previous command, but this time sort the variable by calling the sort() function and supplying that variable. Remember to **choose the variable with the highest mean**.
6. Now repeat the previous command, but instead of sort(), use order(). Remember to **choose the variable with the highest mean**.
7. Write a comment in your R code explaining the difference between sort() and order(). Be prepared to explain this difference to the class.
8. Now use the order command to reorder the whole data frame, store the new dataframe in a variable called ‘sortedDF’
9. Finally, use View() to examine your reordered data frame and be prepared to report on the first few rows

End breakout 1

Breakout 2

1. Get an explanation of the scale() command with the help function:  
   *help("scale")*
2. Run *scale(myIris$Sepal.Length, center=0, scale=T)* and explain the result in a comment. Make sure that all of the members of your group understand what the comment is saying. I may call on anybody!
3. Add a new variable to the myIris data set that contains a scaled version of Sepal.Length, called *scaledSL*

1. Repeat the previous process to add a scaled version of **each of the other three numeric variables**to your data set. Check your results by running the *View()* command on the data set.
2. Add together the four new scaled variables (use + and name each variable individually) and store the result in a new variable called *myIris$scaledSum*.
3. Reorder the myIris data frame based on the values stored in *myIris$scaledSum*. Run *View()* on the result to check your work.