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. Explore myIris via str and the glimpse functions (note: you need to install and library 'tidyverse' to use glimpse). Which do you think is better? Why?

4. Summarize the variables in your copy of the data set, like this:
   summary(myIris)

5. 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

6. Now sort that attribute by calling the sort() function and supplying that variable. Remember to **choose the variable with the highest mean**.

7. Now repeat the previous command, but this time use the order() function, again using **the variable with the highest mean**.

8. Write a comment in your R code explaining the difference between sort() and order(). Be prepared to explain this difference to the class.

9. Now use the order command to reorder the whole data frame, store the new dataframe in a variable called 'sortedDF'

10. Now sort the dataframe using arrange(), which is part of the tidyverse package. This time, **sort based the attribute with the lowest mean.** Store the new dataframe in a variable called 'sortedDF1'

11. Finally, use View() to examine your reordered data frames and be prepared to report on the first few rows

    End breakout 1

Breakout 2

12. What does the following line of code do?
    myIris [, 1]

13. What is the difference (if any) between
    myIris [, "*Sepal.Length*"]
    myIris $*Sepal.Length*

14. Write the R code that outputs the '*Sepal.Length'* attribute values, using the select() command.

15. Create a new column (called 'Ave.Length') in myIris, which, for each row, is the average of Sepal.Length and Petal.Length.

16. What does the following line of code do:
    which.min(myIris$Petal.Length)

17. Using the code from the previous step, output the row (iris observation) with the smallest petal length.

18. Create a new dataframe, with just the Petal.Length and Petal.Width attributes

19. Create a new dataframe, using the slice() function, with only the first three rows in the myIris dataframe.

20. Create a new dataframe, which is a subset of myIris, that only includes rows where Petal.Length is less than 1.4, store in shortPetalDF

21. How many rows are in the shortPetalDF?

22. The homework asks you to create a conditional statement with if and else. A conditional statement is part of a larger group of specialized commands that control the "flow" of a program – what command gets run and when. You can get help on if, else, and other control words. Add and run these commands:
    help("if")
    help("Control")

    Now add and run your first conditional statement:
    if (sum(myNumbers) > 40) print("The sum is greater than 40.")