IST687 – Lab – Storage Wars

Chapter 11, "Storage Wars," describes a variety of ways in which R can connect to data sources. Given that SQL – structured query language – is one of the most fundamental and widely used tools for manipulating data, understand how to use SQL in the context of R is very important. One of the basic data building blocks in R is the data frame and this object bears a very strong resemblance to the concept of a table in SQL. In fact, there is a package in R called "sqldf" that allows for the manipulation of a data frame as an SQL table. This feat is accomplished thanks to SQLite (http://www.sqlite.org), a fantastic, lightweight, open source implementation of SQL. Working with sqldf and SQLite is so convenient that under normal circumstances you can do your work completely within R, with no software installations needed on your computer (other than running the install.packages() command in R).

As we are about halfway through the course, this activity description does not provide the same level of code prompts as previous labs – it is assumed that you remember or can lookup the necessary code. The overall goal of this activity is to use SQL to produce a subset of the built-in "airquality" R dataset that contains only those records where the concentration of ozone is higher than the mean level of ozone. These are the conceptual steps you will need to follow[1]:

1. Review online documentation for sqldf so that you are familiar with the basic concepts and usage of the package and its commands.
2. Install and activate ("library()") the sqldf package in R-Studio. With any new package it is possible to run into installation issues depending on your platform and the versions of software you are running, so monitor your diagnostic messages carefully.
3. Make sure the built-in "airquality" dataset is available for use in subsequent commands. It would be wise to reveal the first few records of airquality with head() to make sure that airquality is available. This will also show you the names of the columns of the airquality dataframe which you will need to use in SQL commands.
4. Using sqldf(), run an SQL select command that calculates the average level of ozone across all records. Assign the resulting value into a variable and print it out in the console.
5. Again using sqldf(), run another SQL command that selects all of the records from airquality where the value of Ozone is higher than the average. Note that it is possible to combine steps 4 and 5 into a single SQL command – those who are familiar with SQL syntax and usage should attempt to do so.
6. Refine Step 5 to write the result table into a new R data object called "newAQ". Then run a command to reveal what type of object newAQ is, another command to show

[1] Submit the output of your runs. Don't forget that the code file you submit for credit must contain full line-by-line comments as well as at least one block comment at the top describing what is going on. Don't forget to cite your sources if you borrow code fragments from elsewhere.

what its dimensions are (i.e., how many rows and columns), and a head() command to show the first few rows.

7. Repeat steps 4,5 and 6 using more "R" like way to do the analysis

Learning Goals for this activity:

A. Refresh or build knowledge of essential aspects of structured query language.
B. Understand the general aspects of the data architecture of R and how it is possible to run SQL commands "natively" in R.
C. Gain experience using the R 'tapply' function
D. Increase independent skills for finding solutions to software configuration issues.
E. Refresh and extend knowledge of the data frame object within R.
F. Increase familiarity with built-in datasets within R.

Essential Guide for All IST687 Activities (appears at the end of all activity guides)

1. All IST687 activities work on what some people call a "constructivist learning" model. By developing a product on your own, testing it to find flaws, improving it, and comparing your solution to the solutions of other people, you can obtain a deeper understanding of a problem, the tools that might solve that problem, and a range of solutions that those tools may facilitate. The constructivist model only works to the extent that the student/learner has the drive to explore a problem, be frustrated, fail, try again, possibly fail again, and finally push through to a satisfactory level of understanding.
2. Each IST687 activity builds on skills and knowledge developed in the previous activities, so your success across the span of the course depends at each stage on your investment in earlier stages. Take the time to experiment, play, try new things, practice, improve, and learn as much as possible. These investments will pay off later.
3. Using the expertise of others, the Internet, and other sources of information is not only acceptable - it is expected. You must ***always, always, always*** give credit to your sources. For example, if you find a chunk of code from r-bloggers.com that helps you with developing a solution, by all means borrow that chunk of code, but make sure to use a comment in your code to document the source of the borrowed code chunk. The discussion boards in the learning management system have been setup to encourage appropriate sharing of knowledge and wisdom among peers. Feel free to ask a question or pose a solution on these boards.
4. Building on the previous point, when submitting code as your solution to the activity, the comments matter at least as much, if not more than the code itself. A good rule of thumb is that every line of code should have a comment, and every meaningful block of code should be preceded by a comment block that is just about as long as the code itself. As noted above, you can use comments to give proper credit to your sources and you can use comments to identify your submission as your own.