Lab

# Lab 5 – External Data: Using SQL

Chapter 11 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, understanding how to use SQL in the context of R is very important. One of the basic data-building blocks in R is the dataframe, 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 dataframe as an SQL table. This feat is accomplished thanks to SQLite ([www.sqlite.org](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 look up 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]](#footnote-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 RStudio. 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 air quality with head() to make sure that air quality is available. This will also show you the names of the columns of the air quality 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 air quality 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 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 tapply, which is a more “R” like way to do the analysis.

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. [↑](#footnote-ref-1)