**Biostatistics**

**Lab 13**

*Tasks*

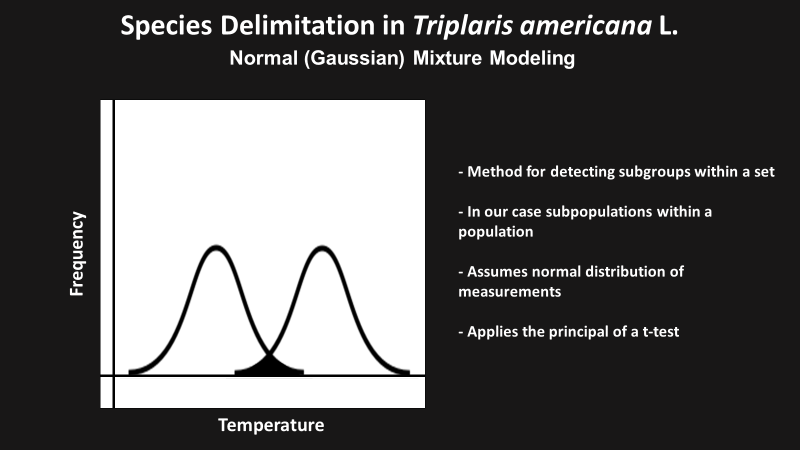
Explore User-Generated R Packages

*Introduction*

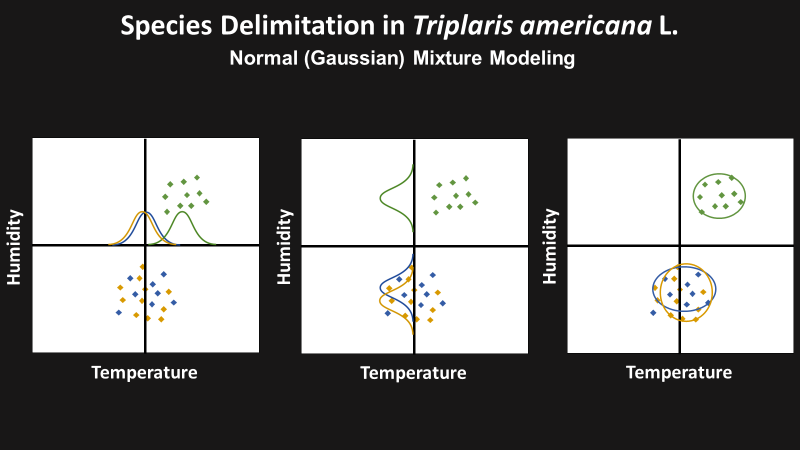
One of the major benefits of R is that it is a regulated open-source program. Anyone who wishes to write their own package or program, may do so. These packages can be evaluated by people associated with the R Project and, if not harmful to people’s computers, can be placed in the CRAN repository or on GitHub.

The purpose of today’s lab is to look at one such package, Mclust. Many of these user-generated packages are not widespread because they were developed for very specific kinds of research. Each program is written by an individual researcher and, though it will follow the rules of R, must be learned on its own terms. The important part is that you have learned the basic skills you need to learn one of these packages.

As states, the package that we will be focusing on is Mclust. Mclust is a Gaussian or Normal Mixture Modeling package. What is Normal Mixture Modeling? A common problem in scientific work is the identification of groups. We know from our studies of the normal distribution and t-tests, that one way to identify groups is by assuming that the observations of suspected groups are normally distributed and then calculating the overlap of those distributions. Because we know a good deal about the normal distribution we can estimate the probability that two suspected groups are actually different from the overlap of their distributions.



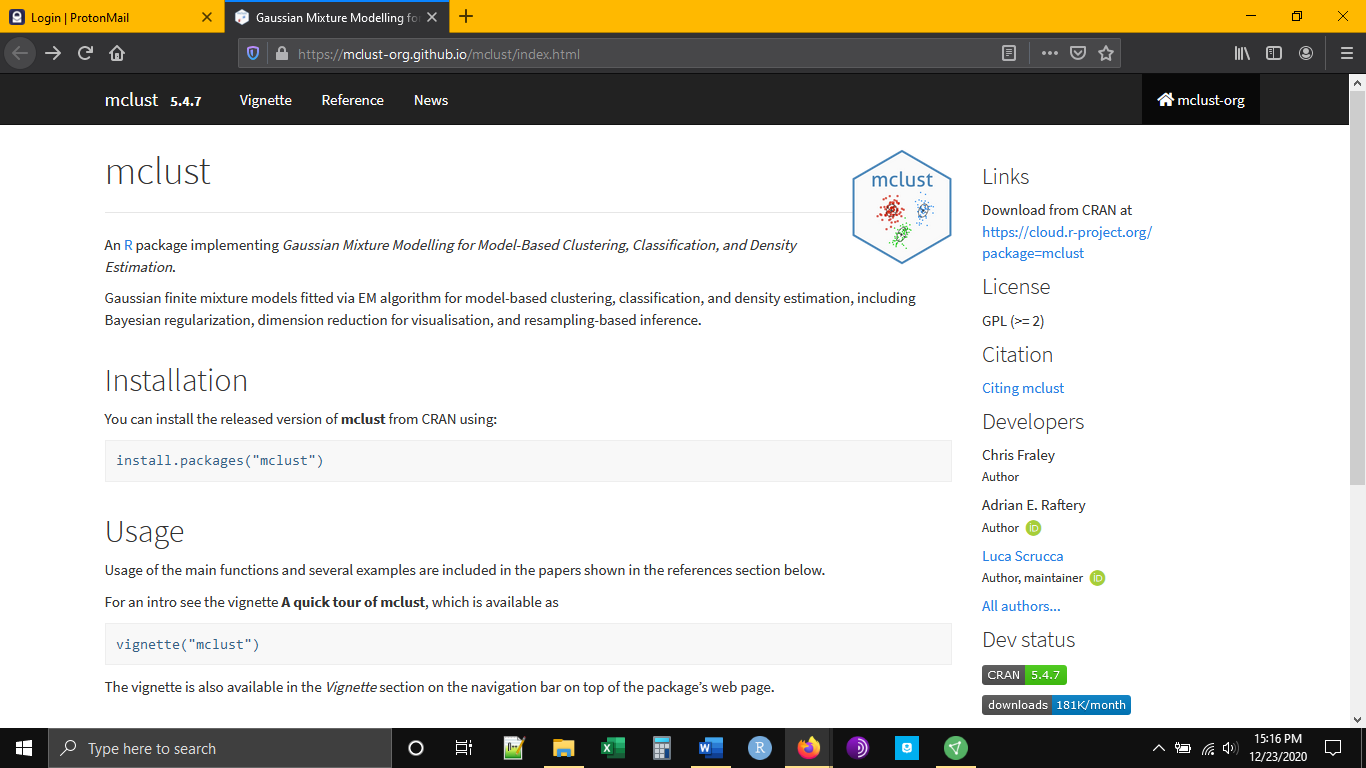
What Normal Mixture Modeling does is apply this same principle to a multivariate problem. For example, let’s say we collected soil samples from three different areas and measured the temperature and the humidity of the soil for each one. We want to know if the soil samples are uniform or if they can be placed into groups. Each field might be a different group. For each variable, Normal Mixture Modeling estimates a series of normal distributions based on the data collected. The overlap of these distributions is calculated for each variable and then synthesized to give a composite probability that observations represent more than one group.



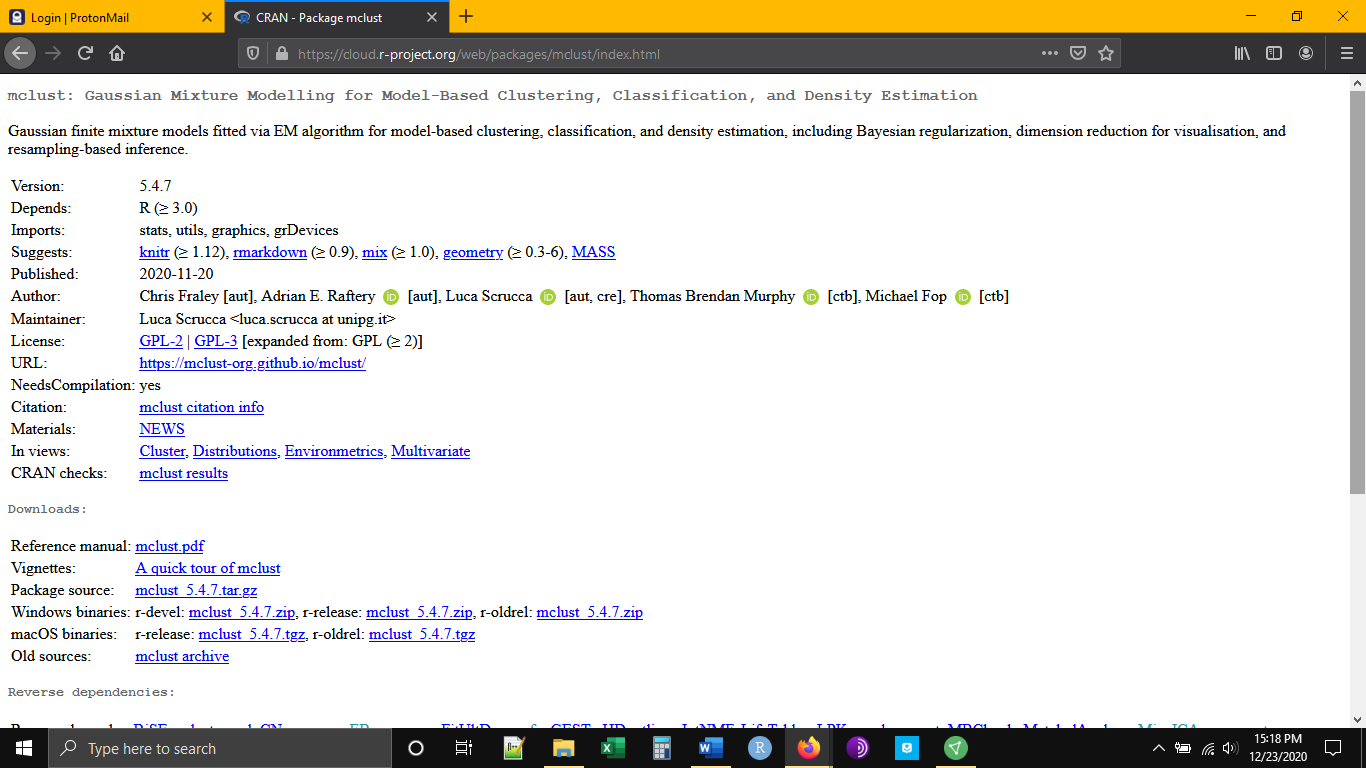
This is what we will be doing today using a *Triplaris* data set we saw earlier in the semester.

*Documentation*

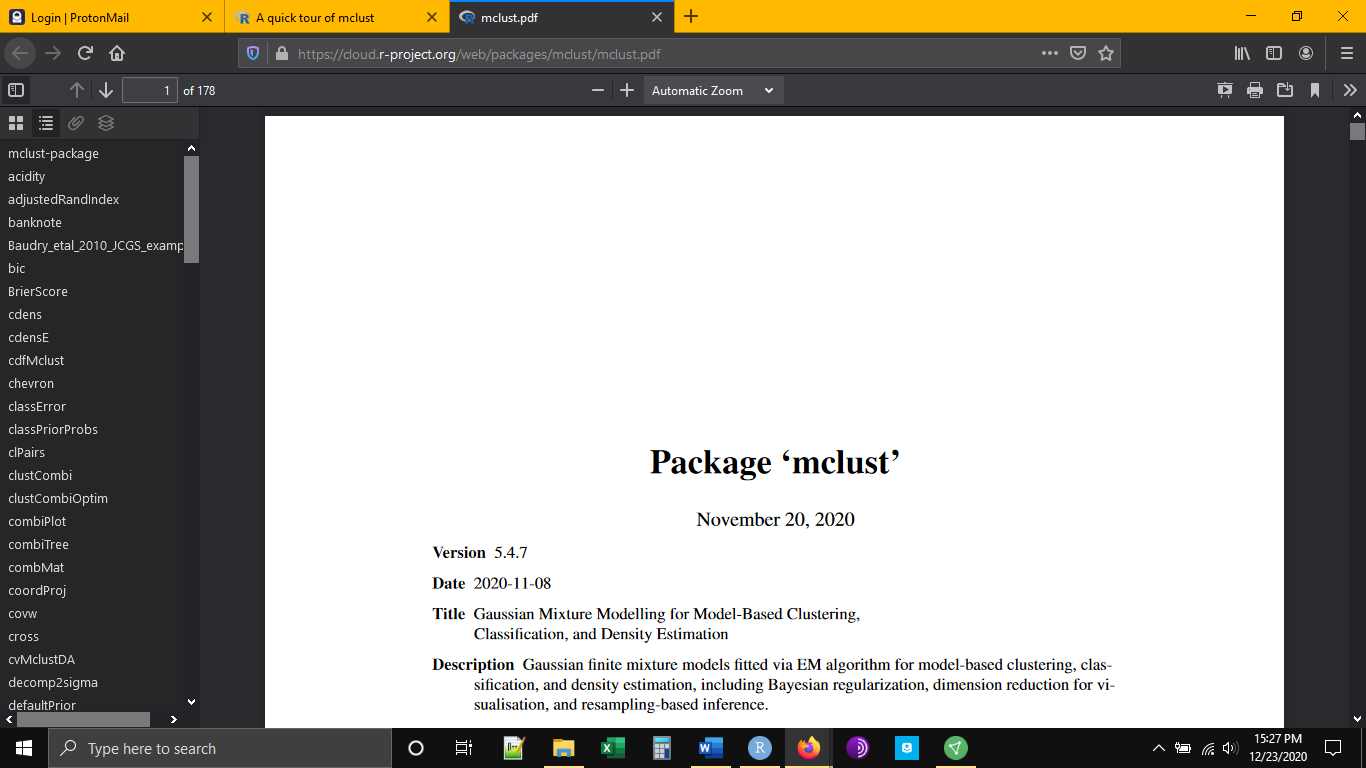
The first thing that we need to do is to download the documentation for Mclust from GitHub and install the Mclust package in R. As indicated elsewhere, GitHub is a safe repository for free computer software. The URL for the Mclust GitHub site is <https://mclust-org.github.io/mclust/index.html>. Once here, we want to click on the link to the CRAN download page (see red box).



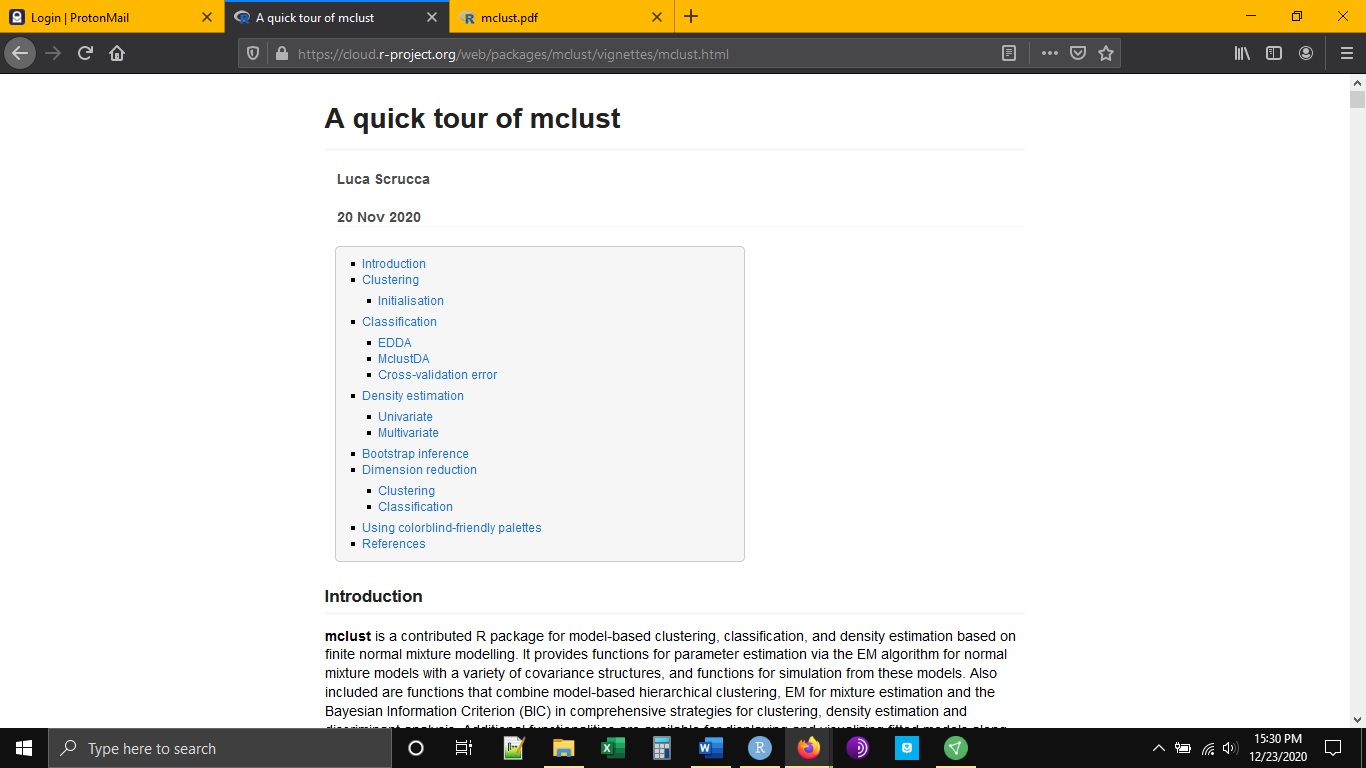
The CRAN page will look as below.



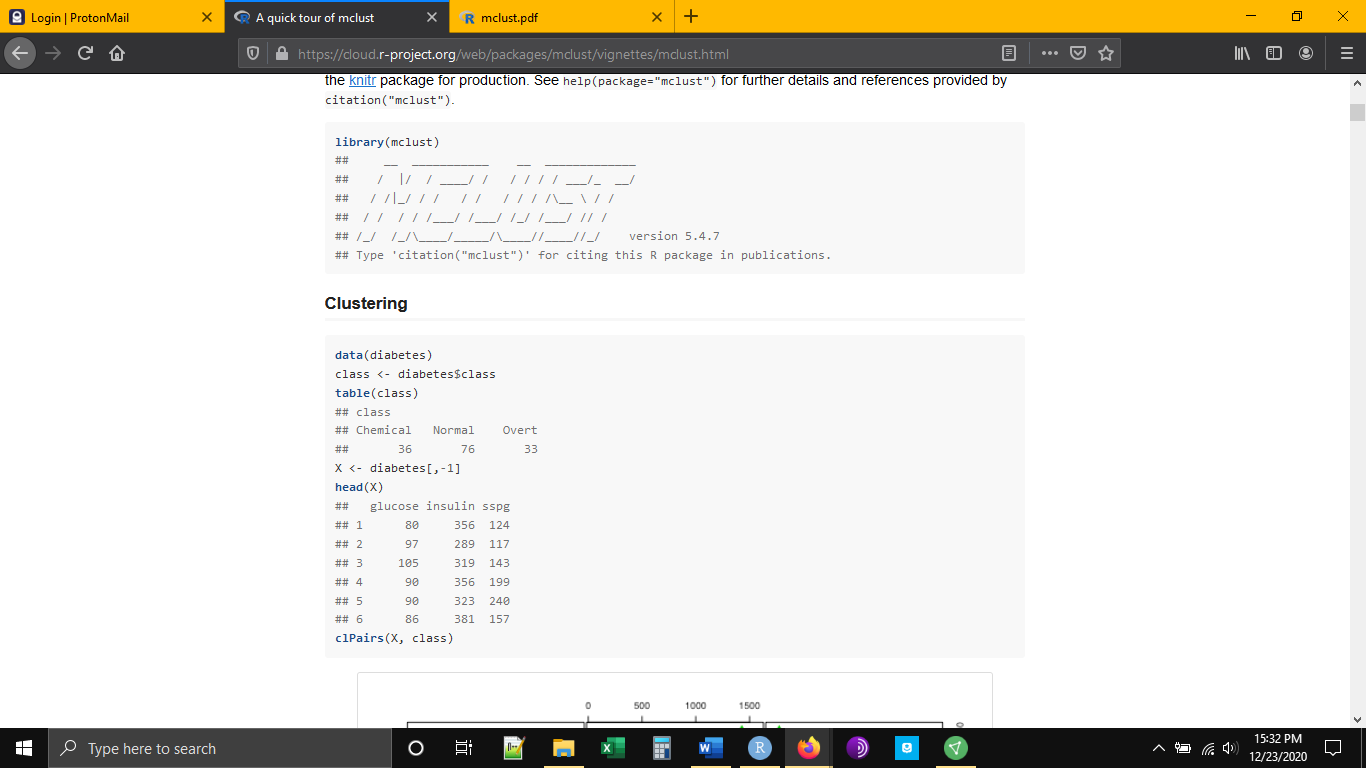
We have two interests here. The first is to download the manual. This will tell us all of the functions inside of the package and what they do.



Our second interest is in an example. Nearly all of the packages available will come with an example. The example is essential because it shows what form the data need to be in before we can run it through the package, as well as the series of functions that will be used within the package to get the desired result. In the case of Mclust the example comes in the form of a “vignette”.



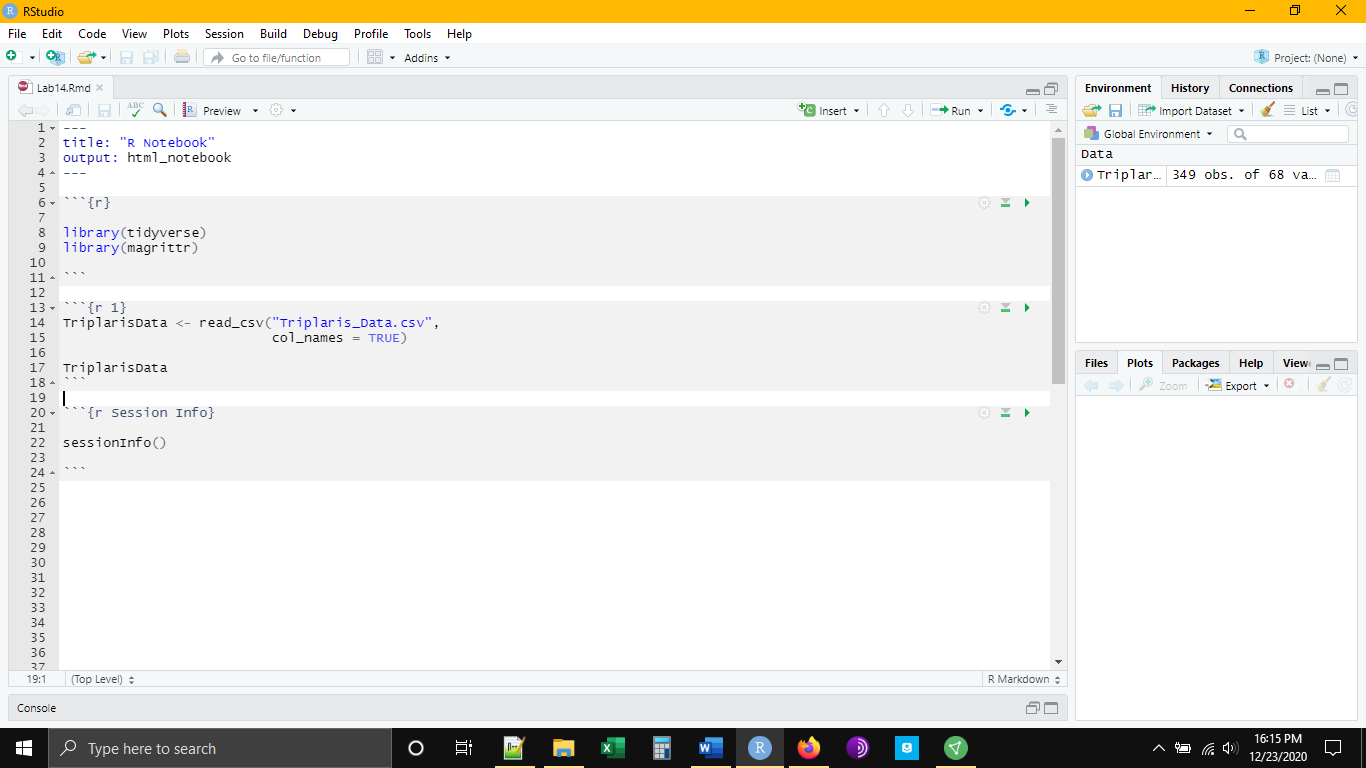
If we scroll down a little we find what we are looking for, an example of the data frame we need to imitate.



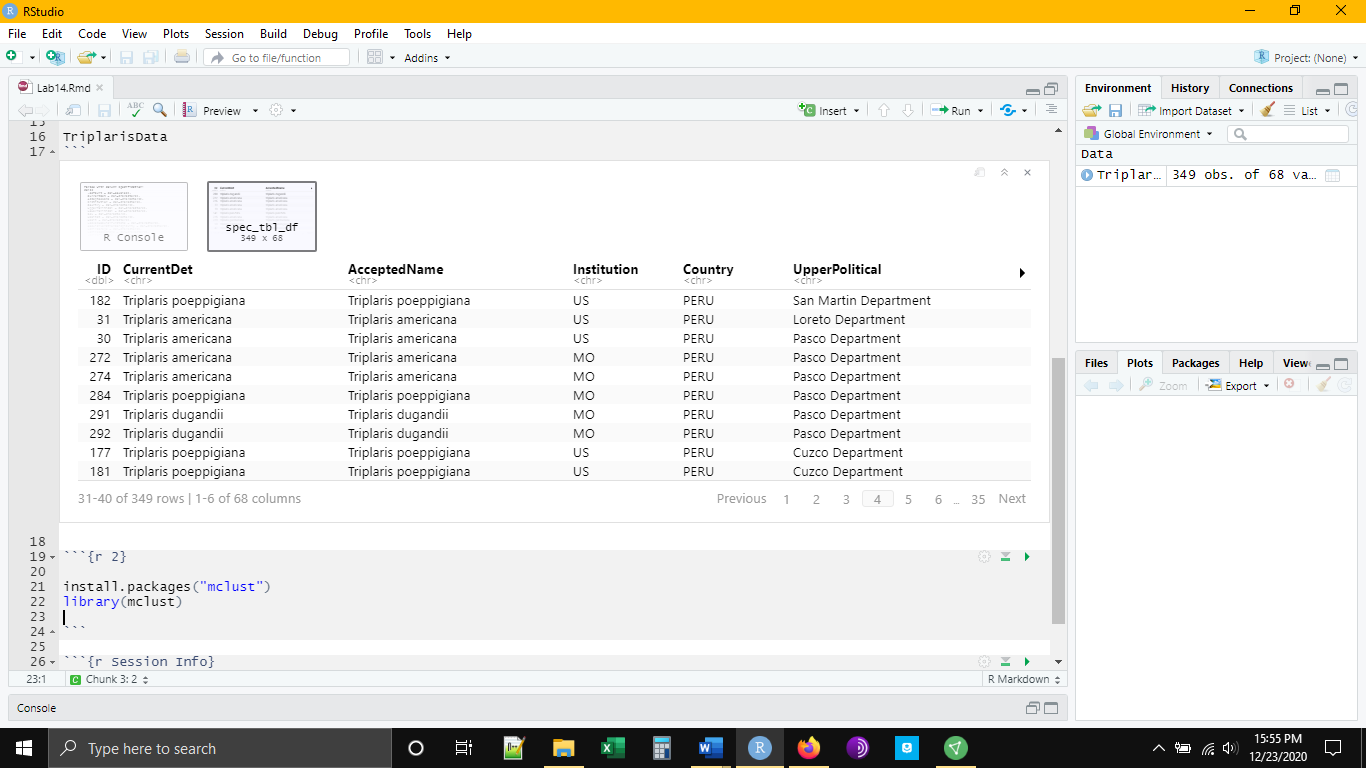
We notice a couple of things. First, all three variables are numeric. Second, we don’t see any missing data. Third, only the observations are present, no vector with putative classes. Our data will need to have these qualities.

*Installation*

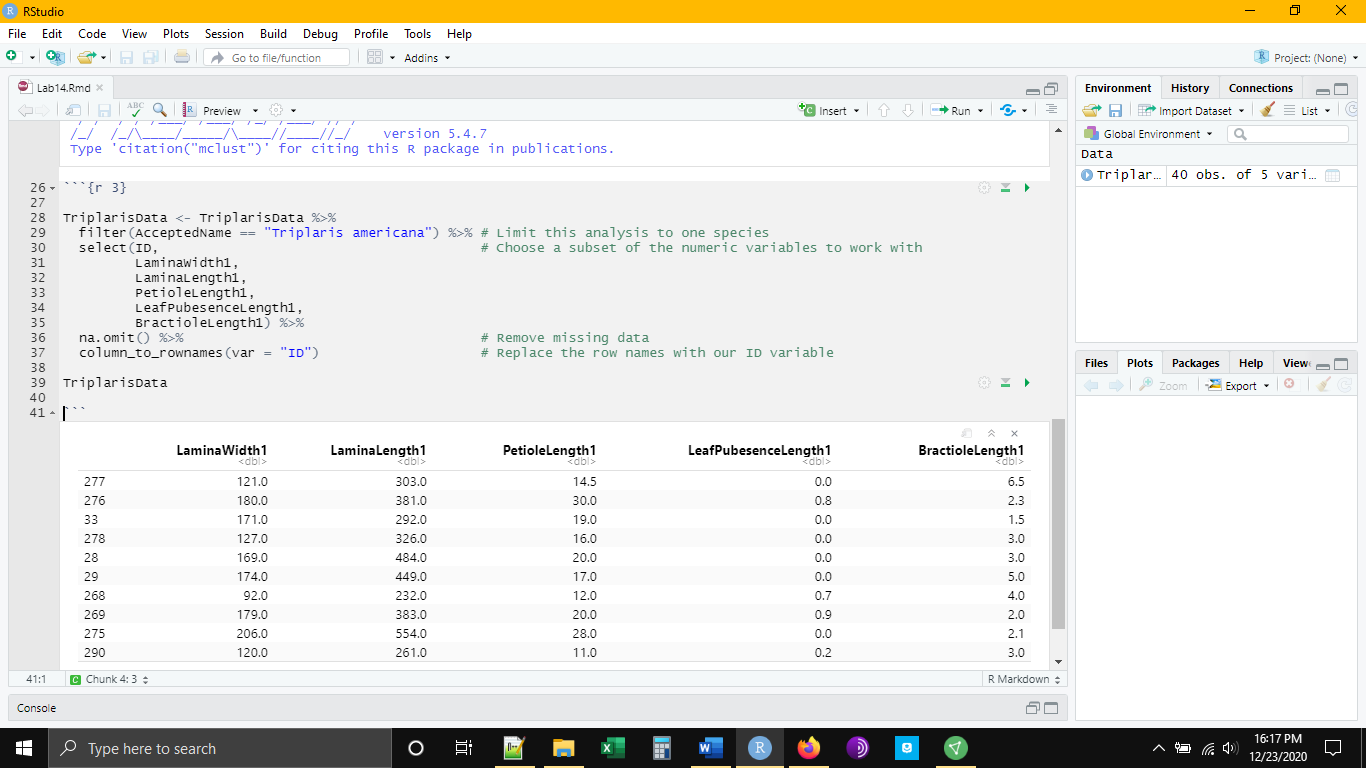
Now it is time to open R Studio and get set up like we usually do. We want to load the tidyverse and read in our data.



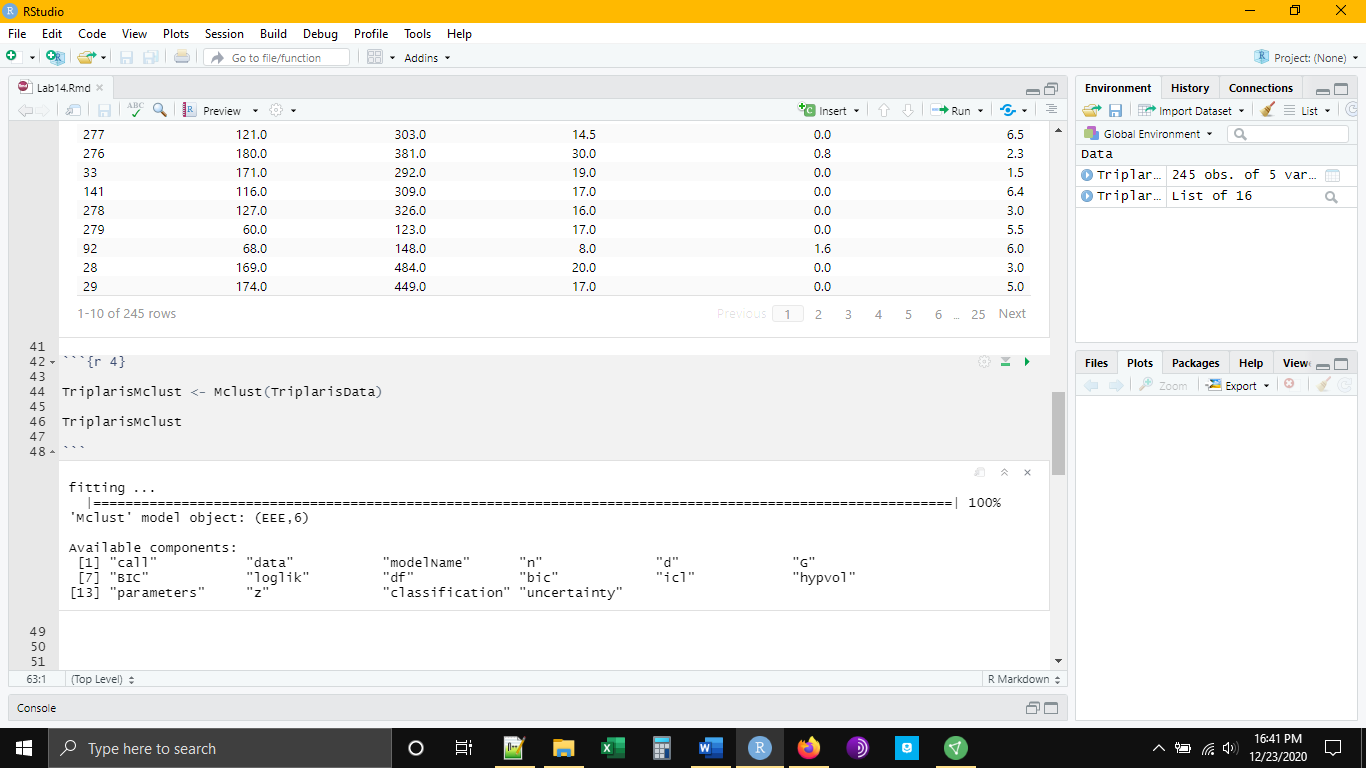
Immediately, we have two tasks. First, we need to install the Mclust package. Second, we need to manipulate our data to put it into the proper form. We can install and load the Mclust package with the following code:



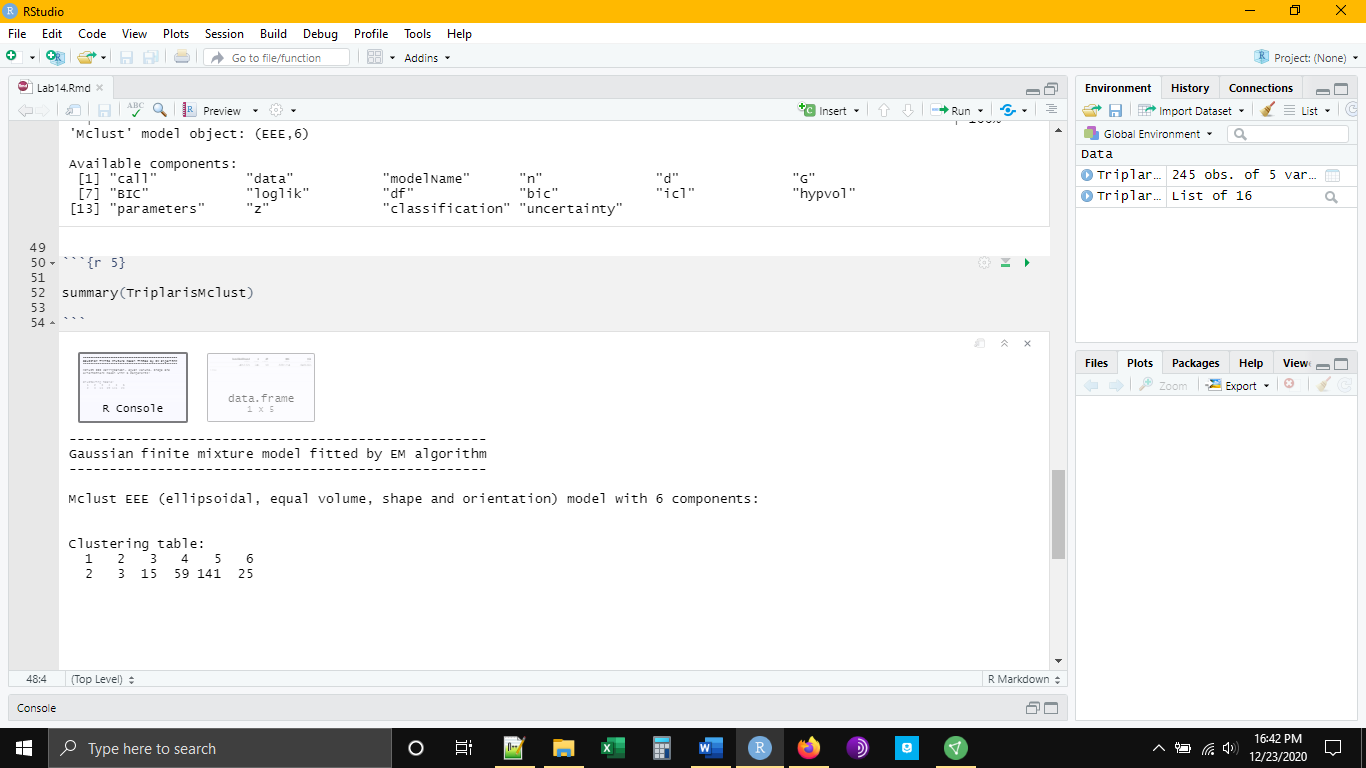
The data present a larger problem. We will need to do a number of things. We will need to remove any categorical variables. We will need to remove any missing data. We will need to move our identifying line. For the sake of simplicity, we will also limit the number of variables present.



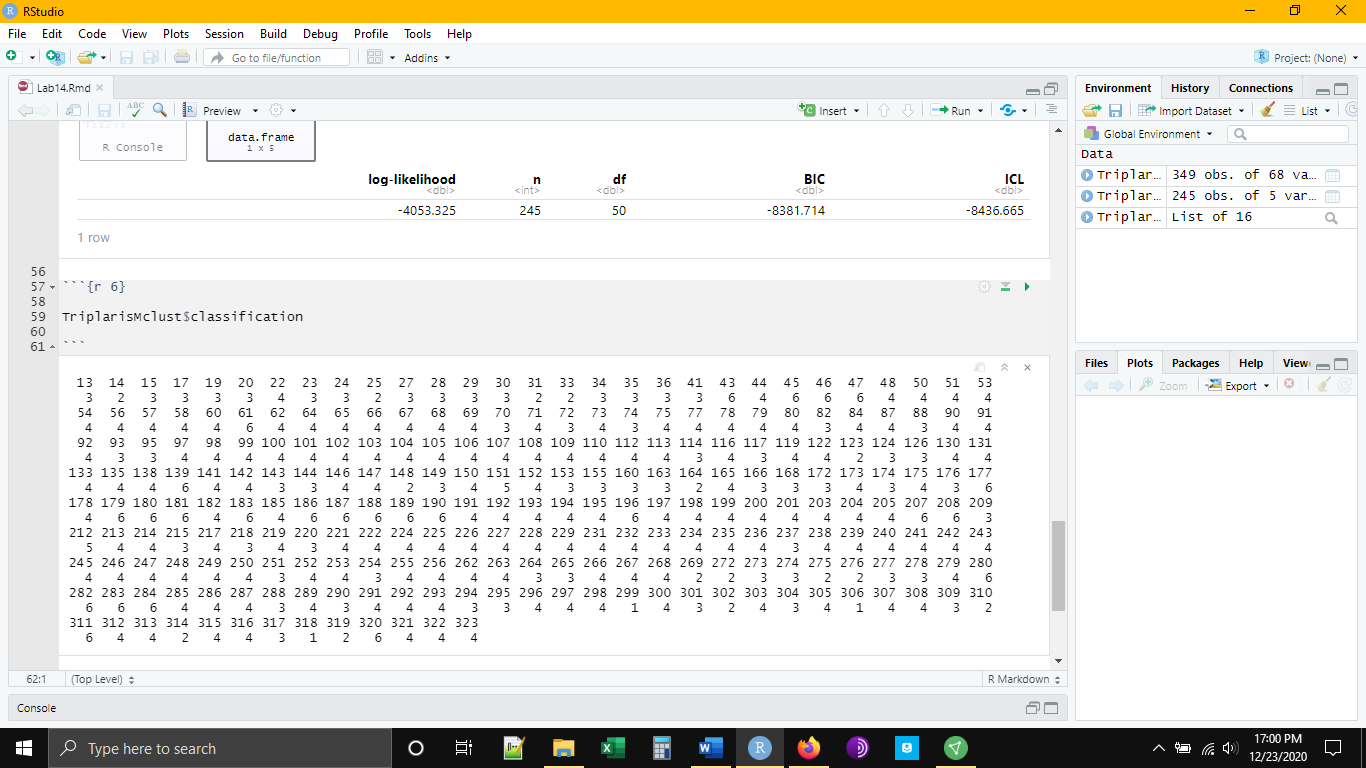
What we have left are 245 observations for five variables. Notice that our ID has now replaced the typical row numbers in the left-hand column. This will help us to track our observations and be able to relate them back to our original data frame if necessary. We are now ready to put our data through Mclust.



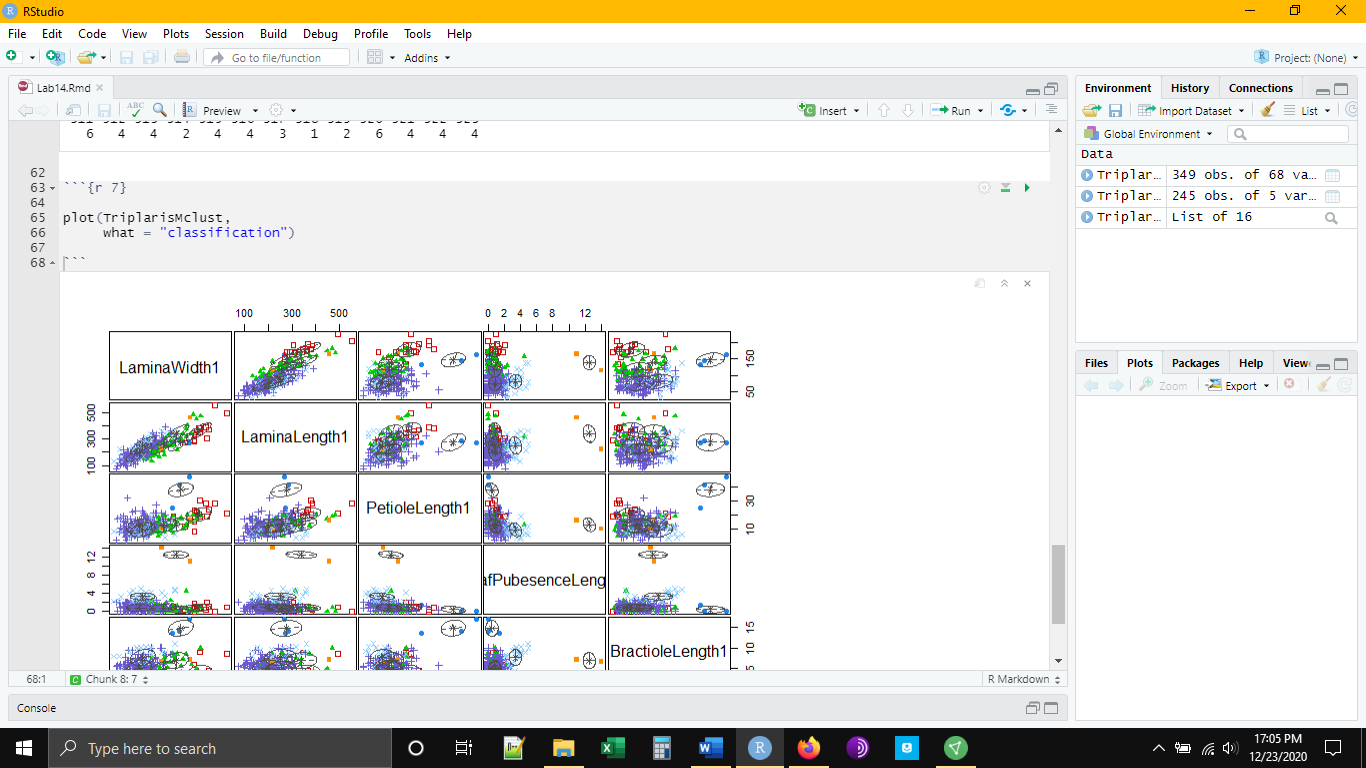
Notice that the output object we saved is really a collection of objects. We can use summary() to get a general sense of our results. We see that Mclust suggests that there are six groups of things in our data set. Below each group it tells how many members are in each group.



If we want to know which observations were assigned to which group, we can ask for the classifications



And, if we want to see the clustering plots used to make those classifications, we can ask for that as well:



*Conclusion*

This finishes our lab for this course. I know that computers can be frustrating at times and learning a new language can be difficult. I sincerely hope that you have benefited from this short introduction to R, R Studio, and the tidyverse and that it will serve as a foundation for future forays in biostatistics.