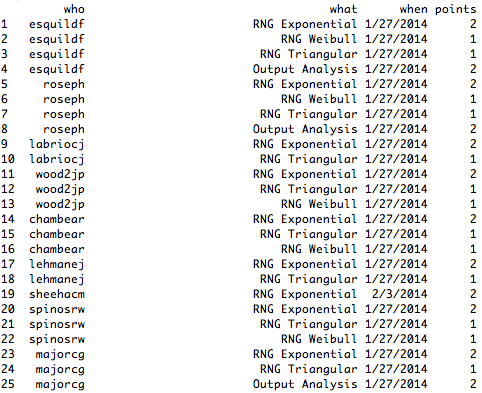
**Data Acquisition: An Introduction to RCurl, File I/O, XML/JSON, and HTML**

In order to be able to mine data, we first and foremost need a source containing some sort of data. In this exercise, we will look into a few ways to access data from different kinds of sources. These include using RCurl to pull data from online sources (such as Google Spreadsheets), reading/writing data from/to a file using File I/O, accessing data in XML/JSON format, and scraping data from an HTML webpage.

**RCurl**

Buried within the list of CRAN packages available to download and utilize in R is a library called RCurl. One of the functions RCurl allows you to do is request the contents of a URL, process the contents in a way that we can mine it, and ultimately store the contents into a variable in R. To help demonstrate this, let’s walk through the example by following the instructions below.

1. Open up an R environment and navigate to the dropdown menu titled Packages (Windows) or Packages & Data (Mac). Click on Install Package(s)… (Windows) or Package Installer (Mac).
2. In the list of libraries, look for a library titled RCurl. Select it and download it. If you are a Mac user, remember to install RCurl’s dependencies by checking the Install Dependencies box.
3. Now that RCurl is downloaded on your computer, lets load it into R. In your R environment, type library(“RCurl”).
4. Now that the RCurl library is loaded, lets utilize it to retrieve some data. In this case, we are going to access some data from a Google Spreadsheet online. To access the data, we are going to perform a few commands simultaneously.
5. In R, type spreadsheet\_data <- data.frame(read.csv(textConnection(getURL("https://docs.google.com/spreadsheet/pub?key=0AoVN55HxlNvKdHh1S2sxQzRkSHlzLW1kVjREVFh4Z2c&single=true&gid=0&output=csv"))))
6. What we did in step five was the following: we used RCurl’s getURL command along with textConnection to establish a connection with the Google Spreadsheet and retrieve its contents. The contents are read in as comma-separated values (csv) using the read.csv command and are then stored within a data frame using the data.frame command. Now, lets make sure that the data has been imported correctly.
7. In R, type spreadsheet\_data. You should see something similar to the image below.



1. The data we just retrieved was from an ISAT 341 class that took place last fall. Each row of the data frame contains a student’s e-ID (who), the name of the assignment completed (what), the date of which the assignment was checked off (when), and the number of points the assignment was worth (points). From here, there are a number of mining operations that we can perform. They include anything from finding out who had the most points to seeing which assignment was most popular among the class.

**File I/O**

Having the ability to read and make additions/edits to a file, (File Input/Output (I/O)) is a very useful tool in that it allows us to make sure a file’s contents are formatted correctly for usage. In the case of this exercise, we will create a simple text document and fill it with some content. Using the scripting language Python, we will read the document’s contents and make some edits to it. You can view this exercise as a Hello World introduction to File I/O