**Semana 1**

**06. Lección de Swirl 4: Leer y Escribir Datos**

**Lección 16 – Leer y Escribir Datos**

Es importante saber que los datos a los que haremos referencia son datos tabulares, es decir, organizados en filas y columnas. Aprenderemos a escribir y leer datos. No todos los datos van a ser escritos en texto, sino que pueden venir en formato de Hojas de Cálculo o Excel e incluso información web. Hay algunas funciones que sirven para leer datos, en forma de archivo de texto, como *read.table, read.csv, readLines, source, dget, load* o *unserialize*. Nos enfocaremos en las dos primeras. Al igual que hay funciones parar leer datos en R, también hay otras para escribirlas, destacando *write.table*.

La función *read.table* es la más usada para leer datos, y en realidad otras funciones tienen parámetros establecidos a partir de ésta. Podemos observar que los parámetros más importantes de esta función son *file, header, sep, colClasses, nrows, comment.char, skip* y *stringsAsFactors*. La función *read.table* es eficiente con archivos pequeños, ya que puede hacer toda la revisión del archivo de forma rápida y eficiente. Si el tamaño es grande, hay que empezar a usar los parámetros. La función *read.table* ignora todas las líneas con #, descubre cuantas filas hay y te dice los tipos de cada uno de los elementos. Si los archivos son más grandes que la memoria RAM no hay que leerlo con esta función.

El parámetro *col.Classes* recibe en un vector cada uno de los tipos de las columnas cuando estemos leyendo los vectores. Cuando hay tres tipos diferentes en un archivo, va leyendo cada uno de ellos y al final decide decidir. Para averiguar la clase de cada columna, se usa el parámetro *class*.

El parámetro *nrows* puede indicar a *read.Table* puede indicar el número de filas que necesita y va a leer y no malgastar tiempo. Es importante saber el sistema que tenemos, es decir, que tenemos que saber el tamaño de la memoria, las aplicaciones en uso, los usuarios que puede haber al mismo tiempo, el sistema operativo y si es de 32 o 64 bits.

La función *write.table*  es similar a la anterior. Para terminar esta función, las funciones para escribir objetos son las funciones *dump y source*. La función *dump* puedes indicar el archivo destino con el parámetro *file*.

**Tarea RStudio**

El argumento *fileEncoding* permite importar el archiv, ya que si hay una segunda cadena que read.table no puede usar no puede hacerlo. Si el programa detecta valores no permitidos como la ñ, hay que introducir la función *ok()*.