

# Cargando datos

*Carlos Iván Espinosa*

*7 de octubre de 2016*

## Contents

<b>Creando un proyecto en RStudio</b>	<b>1</b>
Conozcamos rStudio antes de empezar . . . . .	1
<b>Leyendo Datos</b>	<b>1</b>
Usando read.table . . . . .	1
Usando read_excel . . . . .	2
Verificando los datos . . . . .	3
<b>Ejercicio 1.</b>	<b>5</b>

## Creando un proyecto en RStudio

---

### ¿Cómo creamos un proyecto?

Veamos el siguiente video a ver si nos ayuda.

### Conozcamos rStudio antes de empezar

## Leyendo Datos

---

Vamos a leer unos datos almacenados en formato csv. Existen varias formas para leer datos desde un archivo txt, csv o xls. Empezaremos con los primeros formatos. Antes de nada necesitamos los datos con los que vamos a trabajar los podemos encontrar aquí en formato csv y aquí en formato xls. Descargamos los archivos y los ponemos en la carpeta del proyecto. Para descargar los archivos csv hacer clic en **RAW**

1. El primer paso es saber si nuestro archivo esta en la ubicación del directorio de mi proyecto, para esto utilizaremos la función **dir()**

Teclea en tu consola esta función y mira si tus datos se encuentran ahí?

2. Ahora a cargar los datos usando las funciones `read.table()` y `read_excel()`

### Usando read.table

```
ameLoja<-read.table("AMEBIASIS_LOJA.csv", header=TRUE, sep=';')
```

Si en la consola no ha salido ningun error eso quiere decir que los datos han sido cargados correctamente. Antes de continuar supongo que hay algunas dudas con el código que acabamos de subir.

*¿Que significa header=TRUE?*

Bueno lo que le estamos diciendo es que la primera fila se encuentran los nombres de las variables.

*\*Y sep?*

En este caso, estamos diciendo que separa la separación entre columnas es una coma. Al ser formato csv esto es evidente, pero podría tener un txt separado por tabulaciones por ejemplo, o por cualquier otro caracter.

## Usando read\_excel

```
require(readxl)
```

```
## Loading required package: readxl
```

```
read_excel("AMEBIASIS_LOJA.xlsx",  
           sheet = 1, na = "NA")
```

```
## # A tibble: 3,019 × 9  
##   Cantón Distrito `Dis Distribucion` Sexo `Edad en años`  
##   <chr>    <chr>          <chr> <chr>          <dbl>  
## 1   LOJA    11D01             LOJA Hombre           1  
## 2   LOJA    11D01             LOJA Hombre          13  
## 3   LOJA    11D01             LOJA Hombre          14  
## 4   LOJA    11D01             LOJA Hombre           2  
## 5   LOJA    11D01             LOJA Hombre           2  
## 6   LOJA    11D01             LOJA Hombre          22  
## 7   LOJA    11D01             LOJA Hombre           3  
## 8   LOJA    11D01             LOJA Hombre          30  
## 9   LOJA    11D01             LOJA Hombre          36  
## 10  LOJA    11D01             LOJA Hombre           4  
## # ... with 3,009 more rows, and 4 more variables: `N X` <chr>, `N  
## #   Y` <chr>, Consultas <dbl>, Parroquia <chr>
```

*¿Qué paso?*

La función se ejecuto pero solo se escribió en la consola, claro olvidamos *asignar* estos datos a un objeto, para *asignar* usamos <- y antes de esto ponemos el nombre del objeto.

```
library(readxl)
```

```
ameLojaE<-read_excel("AMEBIASIS_LOJA.xlsx",  
                    sheet = 1, na = "NA")
```

Le he puesto a este objeto al final *E* para saber que es la tabla que he abierto desde excel, si no ponemos esto el objeto que creamos antes será sobrescrito, y r no nos avisará que se sobrescribe así que hay que tener cuidado.

Cuando leemos desde un archivo excel lo primero que debemos hacer es llamar al paquete que nos permite leer archivos excel `'readxl'` esto lo hacemos con la función *library* podríamos utilizar tambien la función *require*.

Lo siguiente es escribir el nombre del archivo, recuerde que R es sensible a las mayusculas así que debe poner exactamente como lo muestra el nombre de su archivo. Posteriormente, le decimos en que hoja se encuentran los datos *sheet* y que debe poner en las celdas vacias *na*, en este caso le decimos que ponga "NA"

## Verificando los datos

Siempre cuando cargo unos datos es necesario asegurarnos que los datos estan bien subidos, para esto utilizaremos dos funciones **head** y **str**

```
head(ameLoja)
```

```
##   Cantón Distrito Dis.Distribucion  Sexo Edad.en.años      N.X
## 1   LOJA   11D01             LOJA Hombre      1 683.887.999.999.509
## 2   LOJA   11D01             LOJA Hombre     13 683.887.999.999.509
## 3   LOJA   11D01             LOJA Hombre     14 683.887.999.999.509
## 4   LOJA   11D01             LOJA Hombre      2 683.887.999.999.509
## 5   LOJA   11D01             LOJA Hombre      2  68.989.299.999.942
## 6   LOJA   11D01             LOJA Hombre     22 683.887.999.999.509
##                                     N.Y Consultas      Parroquia
## 1 957.489.600.000.001      1      CHUQUIRIBAMBA
## 2 957.489.600.000.001      2      CHUQUIRIBAMBA
## 3 957.489.600.000.001      1      CHUQUIRIBAMBA
## 4 957.489.600.000.001      1      CHUQUIRIBAMBA
## 5 956.987.200.000.001      1 TAQUIL (MIGUEL RIOFRÍO)
## 6 957.489.600.000.001      1      CHUQUIRIBAMBA
```

Como ven esta función lo que hace es mostrarnos los seis primeros datos de cada columna. Esto es muy importante para chequear que no se haya cargado los datos de forma erronea. Ahora veamos lo que hace **str**

```
str(ameLoja)
```

```
## 'data.frame':   3019 obs. of  9 variables:
## $ Cantón      : Factor w/ 16 levels "CALVAS","CATAMAYO",...: 7 7 7 7 7 7 7 7 7 ...
## $ Distrito    : Factor w/ 9 levels "11D01","11D02",...: 1 1 1 1 1 1 1 1 1 ...
## $ Dis.Distribucion: Factor w/ 9 levels "CALVAS,GONZANAMA,QUILANGA",...: 5 5 5 5 5 5 5 5 5 ...
## $ Sexo        : Factor w/ 2 levels "Hombre","Mujer": 1 1 1 1 1 1 1 1 1 ...
## $ Edad.en.años  : int  1 13 14 2 2 22 3 30 36 4 ...
## $ N.X          : Factor w/ 94 levels "560.125.999.999.995",...: 56 56 56 56 50 56 56 56 50 56 ...
## $ N.Y          : Factor w/ 94 levels "948.835.300.000.001",...: 73 73 73 73 62 73 73 73 62 73 ...
## $ Consultas    : int  1 2 1 1 1 1 2 1 1 1 ...
## $ Parroquia    : Factor w/ 62 levels "12 DE DICIEMBRE (CAB.EN ACHIOTES)",...: 15 15 15 15 56 15 1
```

Esta función nos permite saber las características del objeto y luego, de cada una de las variables. Como vemos tenemos diferentes tipos de variables algunas son categóricas y otras numéricas.

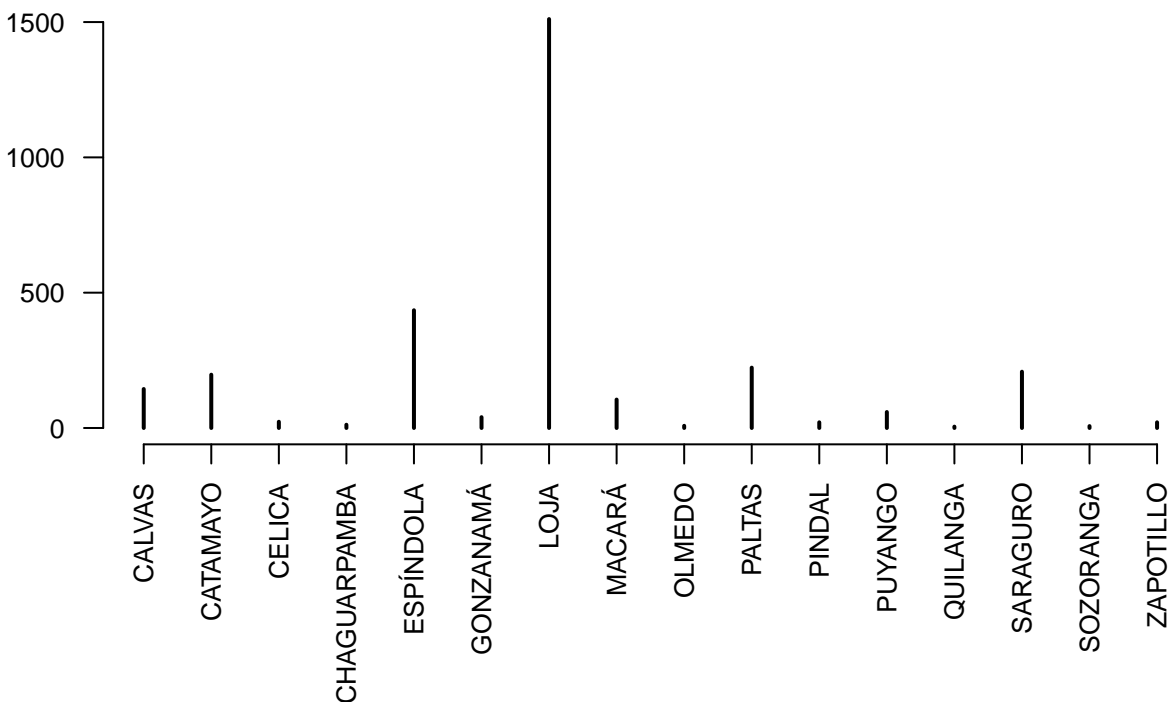
Seguramente se estarán preguntando que son estos datos, bueno estos datos corresponden a estadísticas de amebiasis en la provincia de Loja. Vamos a realizar un primer gráfico para conocer como la insidencia de amebiasis se distribuye en los diferentes cantones.

```
table(ameLoja$Cantón)
```

```
##
##      CALVAS      CATAMAYO      CELICA CHAGUARPAMBA      ESPÍNDOLA
##      144        197         23         12         435
##  GONZANAMÁ      LOJA      MACARÁ      OLMEDO      PALTAS
##      40        1511        105         8        223
##      PINDAL      PUYANGO      QUILANGA      SARAGURO      SOZORANGA
##      21         59         5        208         7
##  ZAPOTILLO
##      21
```

Creo que sería mejor verlo en un gráfico, lo que hacemos es poner plot al principio de la función table.

```
par(mar=c(9,3,2,1))
plot(table(ameLoja$Cantón), las=2, cex.axis=0.8)
```



Como podemos ver en Loja la cantidad de Amebiasis es mucho mayor que en los otros cantones.

Pero, les parece que esta conclusión de que en el Cantón Loja hay más amebiasis es correcta.

Bueno les dejo con la duda, o mejor con el trabajo.

Ustedes deben trabajar con estos datos y decir si esta conclusión es cierta.

## Ejercicio 1.

---

Seguramente seguirán preguntándose si el gráfico de amebiasis es correcto o no (*eso espero*).

¿Qué realmente me esta diciendo el gráfico?

Lo que realmente me dice, es la cantidad de amebiasis en cada cantón, no la incidencia de amebiasis. Vamos a trabajar. Contesten las siguientes preguntas y desarrollen los ejercicios.

1. Busque en internet la definición de insidencia y corrija los datos que hemos obtenido. Para esto debemos *asignar* la tabla en un objeto, espero recuerde como hacerlo.
2. Queremos graficar estos resultados, pero que se grafique de una forma ordenada del cantón con menos incidencia al cantón con más incidencia. Para esto utilizaremos la función **order**. Use `?` o `help()` para saber como puede utilizar la función `order`.

Con esto hemos terminado el primer ejercicio. Espero que esten satisfechos ya estan trabajando en **r** han aprendido como cargar los datos usando `read.table` y `read_xls`, han visto como podemos utilizar funciones como; `order`, `head` y `str`, y hemos iniciado a trabajar con gráficas. En las siguientes lecciones nos adentraremos en funciones que nos servirán para conocer nuestros datos.

