UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE MATEMÁTICA


Licenciatura en Estadística

# Control Estadístico del Paquete R

## "UNIDAD UNO"

Alumna:
Erika Beatríz Guillén Pineda

Fecha de elaboración
Santa Ana - 27 de noviembre de 2015

# 1. IMPORTACIÓN Y EXPORTACIÓN DE DATOS EN R

Generalmente los datos suelen leerse desde archivos externos y no teclearse desde la consola. Las capacidades de lectura de archivos de R son sencillas y sus requisitos son bastante estrictos, por lo que hay que tenerlas muy en cuenta, de lo contrario los resultados en la lectura no serán los esperados.

## 1.1. USO DE LA FUNCIÓN READ.TABLE()

- Ejemplo: Guardar (escribir) determinados datos en un archivo de texto (ASCII) y luego recuperar (leer) dicho archivo desde R.
- 1) Cambiar el directorio de trabajo a su directorio de trabajo, en el cual ha almacenado sus pr?cticas, desde el menú File.
- 2) Abrir el R Editor para crear un nuevo script desde el menú File.
- 3) En la ventana del R Editor, teclee los datos tal como se muestra:

Observaciones:

- La primera línea del archivo debe contener el nombre de cada objeto o variable.
- En cada una de las siguientes l?neas, el primer elemento es la etiqueta de la fila, y a continuación deben aparecer los valores de cada variable.
- Si el archivo tiene un elemento menos en la primera línea que en las restantes, obligatoriamente seí el diseño anterior el que se utilice.
- A menudo no se dispone de etiquetas de filas. En ese caso, también es posible la lectura y el programa a´ nadir? unas etiquetas predeterminadas.
- La última línea debe finalizar con ENTER para que R reconozca el fin del archivo.
- 4) Oprimir con el puntero del ratón el icono que representa un disquete (Save script as) y guarde el archivo con el nombre "datos01.txt". También puede darle el nombre de "datos01.dat"(otro formato soportado por la función read.table), e incluso puede leer datos directamente desde una página de internet, solamente proporcionando la dirección URL completa

- 5) Recuperar los objetos o datos guardados en el archivo "datos01.txt"

```
Entrada1 <- read.table("datos01.txt", header=T);Entrada1

##   Edad Estatura Peso Sexo
## 1   26     1.65  146    F
## 2   21     1.73  158    M
## 3   21     1.81  167    M
## 4   20     1.70  152    F

Entrada2 <- read.table("datos01.dat", header=T);Entrada2

##   Edad Estatura Peso Sexo
## 1   26     1.65  146    F
## 2   21     1.73  158    M
## 3   21     1.81  167    M
## 4   20     1.70  152    F

# No existe diferencia entre ambos archivos a la hora de leerlos
```

NOTA: La función read.table() lee los datos y los almacena en una hoja de datos (data.frame), si búsqueda.

- 6) Leer los datos contenidos en el archivo .ªirline.dat"

```
airline <- read.table("airline.dat",nrows = 20);airline

##        V1
## 1  112,00
## 2  115,00
## 3  145,00
## 4  171,00
## 5  196,00
## 6  204,00
## 7  242,00
## 8  284,00
## 9  315,00
## 10 340,00
## 11 360,00
## 12 417,00
## 13 118,00
## 14 126,00
## 15 150,00
## 16 180,00
## 17 196,00
## 18 188,00
## 19 233,00
## 20 277,00
```

```
# Note que la instrucci\'on header=T es por defecto y puede omitirla (R reconocer?
# siempre que en la primera l?nea se encuentran los nombres de las variables).
```

La sintaxis completa de la función read.table() es:

read.table(file, header = FALSE, sep = , quote = "¨", dec = ".", row.names, col.names, as.is =FALSE, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill =!blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "")

# 2. USO DE LA FUNCIÓN SCAN()

La función scan() es más flexible que read.table() y permite realizar lecturas m?s complejas, como puede consultar en la ayuda: help(scan) itemize

Ejemplo 1: Leer sólo las dos primeros objetos o columnas del archivo "datos01.txt"

```
Edat1 <- scan("datos01.txt", list(X1=0, X2=0), skip = 1, flush = TRUE, quiet = TRUE);
Edat1

## $X1
## [1] 26 21 21 20
##
## $X2
## [1] 1.65 1.73 1.81 1.70

Edat2<- scan("datos01.dat", list(X1=0, X2=0), skip = 1, flush = TRUE, quiet = TRUE);
Edat2

## $X1
## [1] 26 21 21 20
##
## $X2
## [1] 1.65 1.73 1.81 1.70

# Observe que en list(X1=0, X2=0) se les da el nombre a las dos primeras
# columnas o variables (puede darle el nombre que crea m\'as conveniente) y se
# indica que son variables num\'ericas; sin embargo, del archivo \'unicamente se
# leen las dos primeras columnas, si se quisiera leer las columnas primera y
# tercera, nos ver\'iamos obligados a leer las tres primeras

# Note que si escribimos list(0, 0), indica que se leer\'an las dos primeras
# columnas del archivos y que los datos le?dos son num\'ericos (asigna nombres
# por defecto). Para indicar que los datos que se leen son cadenas se utiliza
# "" en lugar de 0.
```

- Ejemplo 2: Crear un archivo con la función cat() y luego recuperarlo

```
cat("TITULO L\'inea extra", "2 3 5 7", "11 13 17", file="datos02.txt", sep="\n")

# El archivo lo recuperamos con la funci\'on scan():
pp <- scan("datos02.txt", skip = 1, quiet= TRUE);pp

## [1]  2  3  5  7 11 13 17

# La funci?n scan es muy \'util cuando en el archivo de datos a importar
# cada l?nea representa un ?nico caso. En caso contrario (cada cierta
# cantidad de columnas representa un caso) es mucho m\'as f\'acil y
# recomendable utilizar la funci\'on read.table.
```

La sintaxis completa de la funci?n scan() es:

scan(file = , what = double(0), nmax = -1, n = -1, sep = , quote = if (sep==””) else ¨, dec = ”.”, skip = 0, nlines = 0, na.strings = ”NA”, flush = FALSE, fill = FALSE, strip.white = FALSE, quiet = FALSE, blank.lines.skip = TRUE, multi.line = TRUE, comment.char = ””)

# 3. USO DE LA FUNCIÁN READ.CSV()

Leer un conjunto de datos de Microsoft Excel pero los datos no están almacenados en el formato conocido de Excel ”.xls”, sino más bien un formato menos conocido como ”.csv”.

- 1) Ingresar al Microsoft Excel y crear la hoja de datos siguiente:
  Observe que debe guardar la hoja Excel en su directorio de trabajo y que el archivo debe ser de tipo: CSV(delimitado por comas)
- 2) Regresar al entorno de R y recuperar el archivo ”HojaE1.csv”.

```
hojaR <- read.csv("HojaE1.csv", sep = ";", strip.white = TRUE)
hojaR

##   Producto.Cantidad.S1.Cantidad.S2.Cantidad.S3.Cantidad.S4
## 1                                  Desayuno,132,125,142,120
## 2                                  Almuerzo,120,125,122,114
## 3                                     Cenas,115,105,130,108
## 4                               Tazas de cafÃ©,200,180,210,140
## 5                                     Gaseosas,75,90,62,80
```

**Note que R ha reemplazado .en los encabezados de las columnas por ”.”; en general reemplazará cualquier carácter.**

Puede investigar el tipo de objeto que es hojaR con:

```
is.matrix(hojaR);

## [1] FALSE

is.list(hojaR);

## [1] TRUE

is.data.frame(hojaR)

## [1] TRUE
```

Acceda a la componente Producto de hojaR con:

```
hojaR$Producto

## [1] Desayuno,132,125,142,120      Almuerzo,120,125,122,114
## [3] Cenas,115,105,130,108         Tazas de cafÃ©,200,180,210,140
## [5] Gaseosas,75,90,62,80
## 5 Levels: Almuerzo,120,125,122,114 ... Tazas de cafÃ©,200,180,210,140
```

Observe que R toma está columna (variable de caracteres) como un Factor Nominal, verifíquelo tecleando:

```
is.vector(hojaR$Producto);

## [1] FALSE

is.factor(hojaR$Producto)

## [1] TRUE
```

Qué tipo de objeto es la columna Cantidad.S1?

```
is.vector(hojaR$Cantidad.S1);

## [1] FALSE

is.factor(hojaR$Cantidas.S1)

## [1] FALSE
```

# 4. USO DEL PAQUETE RODBC

Si por el contrario los datos a los cuales deseamos realizar el an?lisis estad?stico se encuentran en formato XLS (versión 2003 de Microsoft Excel), debemos de seguir los siguientes pasos (Ilustraremos el procedimiento con el archivo çontaminación-mexico.xls"):

- Instalar el paquete RODBC, con la siguiente instrucción

```
install.packages(c("RODBC"))


## Installing package into 'C:/Users/User/Documents/R/win-library/3.2'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without setting a
mirror

# O desde el men\'u como en el caso de la instalaci\'on del paquete Foreing.
# Con este procedimiento se instalan los paquetes directamente desde internet,
# es necesario para ello contar con una conexi\'on a internet en el momento.
# Posteriormente se selecciona un mirror (un servidor desde el cual se
# descargar\'an los paquetes), y finalmente buscar el paquete deseado del listado.
```

- Cargar el paquete con la siguiente instrucción:

```
library(RODBC)


## Warning: package 'RODBC' was built under R version 3.2.2
```

- Seleccionar el archivo (el cual puede contener m?s de una hoja de datos)çontaminación-mexico.xls", con la instrucción:

```
datos.xls <- odbcConnectExcel(file.choose())


## Error in odbcConnectExcel(file.choose()): odbcConnectExcel is only usable with
32-bit Windows
```

- Seleccionar la hoja en la cual se encuentran los datos

```
# datoshoja1.xls <- sqlFetch(datos.xls,"contaminacion-mexico")
# Con esta instrucci\'on se indica la hoja en la cual se encuentran los
# datos con los que se desea trabajar (contaminaci?n_mexico)
# o cargar en R. Siempre es necesario especificarlo.
```

- Realizar los análisis o cálculos correspondientes

# 5. IMPORTAR DATOS DE SPSS HACIA R

A parte de leer archivos en formato texto y delimitados por comillas, R permite leer datos en una gran variedad de formato entre ellos se encuentra archivos el formato de SPSS ".sav". Para poder leerlos primero debemos de cargar el paquete correspondiente en el cual se encuentran la función que nos permitir? leer los ficheros de datos. Para el caso de SPSS, debe cargar el paquete foreign. El cual es necesario para lectura y escritura de datos.

Para leer los datos se usa la siguiente función Read.spss("nombreArchivo", use.values.labels.=FALSE, max.value.label=Inf, to.data.frame=T); donde use.values.labels=TRUE significa que si en el archivo existen variables categóricas que han sido previamente codificadas con su respectiva etiqueta, entonces se leerán directamente las etiquetas y no los valores de esta (por ejemplo, si 1 representa Femenino, se leer? Femenino en lugar de 1). to.data.frame =T indica que los datos ser?n almacenados en un data.frame, muy recomendable para an?lisis estadístico. Puede consultar más ayuda de la función con la instrucción help(read.spss).

- Instalar el paquete foreign, con la siguiente instrucción

```
install.packages(c("foreign"))

## Installing package into 'C:/Users/User/Documents/R/win-library/3.2'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without setting a
mirror
```

- Cargar el paquete con la siguiente instrucción:

```
library(foreign)
```

- Leer el contenido del archivo "demo.sav", con la instrucción:

```
#read.spss("demo.sav", use.value.labels=TRUE, max.value.label=Inf, to.data.frame=T)
```

- Realizar los an?lisis o cálculos correspondientes.