CURSO DE R (II). INTRODUCIR Y GUARDAR MIS DATOS EN R

Jon Molinero Ortiz *

Más sobre las funciones de R

Antes de avanzar en las utilidades de R, aprendamos algo más sobre sus funciones. Además, aprenderemos a utilizar la información que nos da R en su manual de ayuda. Busquemos información sobre una función que ya conocemos del capítulo anterior:

> ?log

al presionar ENTER se abrirá el explorador de Internet y mostrará la página de ayuda de la función "log" y otras funciones relacionadas. La página de ayuda comienza con una descripción de las funciones (Description), seguido de otro apartado que muestra cómo se utiliza la función en el editor de comandos de R (Usage). En el caso de la función "log", podemos leer que la función calcula logaritmos y obtenemos la siguiente información de uso:

log (x, base = exp(1))

pero para comprender lo que significa esta información debemos continuar leyendo las siguientes secciones que explican los argumentos (Arguments) y los valores que devuelve la función (Value). La función "log" tiene dos argumentos: el primero se llama "x" y es el número del cual queremos calcular su logaritmo, y el segundo argumento se llama "base" y es la base para calcular el logaritmo. La ayuda también nos informa de que el valor por defecto de "base" es "exp(1)", el número e. Esto significa que, si no incluimos un valor para el argumento "base" al usar la función, entonces R le asigna automáticamente el valor de e. Si queremos calcular el logaritmo de 100 en base 10 podemos llamar a la función sin usar los nombres de los argumentos:

> log (100, 10)

o usando los nombres de los argumentos:

> log (x=100, base=10)

al presionar ENTER obtendremos el mismo resultado en ambos casos:

[1] 2

La ventaja de usar los nombres de los argumentos al utilizar las funciones de R es que no es necesario recordar el orden en el que se deben introducir los argumentos en la función. Así, si tecleamos:

> log (base=10, x=100)

la función nos sigue calculando el logaritmo de 100 en base 2. Sin embargo, si tecleamos:

> log (10, 100)

y presionamos ENTER, obtenemos:

[1] 0.5

que es el logaritmo de 10 en base 100. En resumen, si no utilizamos los nombres de los argumentos, tenemos que introducir los argumentos de la función exactamente en el mismo orden que aparece en el manual de R. Si utilizamos los nombres de los argumentos, podemos introducir los argumentos de la función en cualquier orden que queramos. Y esto es muy útil en funciones que tienen muchos argumentos como veremos a continuación.

¿Cómo tabular mis datos para trabajar en R?

Tipos de datos en R

Sabemos que R guarda la información en objetos mediante el operador de asignación "<-":

> b <- 12

Según la necesidad, R puede almacenar diferentes tipos de datos en un objeto (Tabla 1). Utilizaremos objetos de tipo numérico (tipo "numeric") para almacenar variables numéricas o cuantitativas y objetos de tipo texto (tipo "character") para almacenar variables de tipo categórico o cualitativo. El tipo lógico (tipo "logic") sirve para almacenar variables que sólo pueden contener el valor T, de verdadero (en inglés True), o el valor F, de falso (en inglés False).

Tabla 1. Tipos de datos que reconoce R

Tipo de dato	Nombre en R	Ejemplo
Número entero	Numeric	> mi.edad <- 47
Número real	Numeric	> mi.altura <- 1.68
Texto	Character	> mi.nombre <- "Jon Molinero"
Valor lógico	Logic	> soy.soltero <- F

¿Cómo organizo mis datos?

Podemos tabular nuestros datos en EXCEL o tabularlos directamente en R como se explica más adelante. Pero seguramente, al empezar a tabular nos surgirá una pregunta: ¿Cómo tengo que organizar mis datos? ¿Existe alguna regla para introducir mis datos en R? La respuesta puede ser si o no. Digamos que yo os puedo dar una serie de recomendaciones para tabular los datos que hacen el uso de R más fácil, aunque probablemente otra persona puede haber desarrollado otras formas de organizar y usar R.

Antes de empezar a tabular, debemos examinar nuestros datos, aprender a identificar las observaciones y las variables. Las observaciones son cada uno de los objetos de nuestro estudio, ya sean personas, cuadrantes, muestras de bentos o de agua, y las variables son las mediciones que hemos realizado en cada objeto observado. Además, podemos incorporar algunas variables categóricas que nos den información sobre las observaciones como la fecha de muestreo, el lugar, el número de réplica o el nivel de un factor en un experimento entre otras. Si hemos realizado encuestas, las observaciones son las personas encuestadas y las variables son las respuestas al cuestionario de la encuesta. Si se trata de salidas de campo, las observaciones son las muestras recogidas y las variables las medidas que hemos realizado en cada muestra en el campo o en el laboratorio. Si hemos estudiado la abundancia de especies, las observaciones son los cuadrantes o las localidades donde se han realizado los contajes y las variables son las especies encontradas. Si se trata de un experimento de campo o de laboratorio, las observaciones son las unidades experimentales y las variables son las medidas que hemos realizado en ellas. Si se trata de series temporales, como datos climáticos como la temperatura o la precipitación en diferentes localidades, las observaciones son las fechas sucesivas y las variables son las medidas realizadas en cada una de las localidades. A la hora de tabular los datos, las observaciones se tabulan como filas de nuestra tabla de datos y las variables como las columnas.

Existen además otras buenas prácticas a la hora de tabular nuestros datos que harán más fácil el uso de R:

- No debemos dejar celdas en blanco a la izquierda ni por encima de nuestra tabla de datos.

- La primera columna de la tabla debe de ser un indicador único para identificar las observaciones.

- Utilizar letras minúsculas para los nombres de las variables, así "total" es un nombre mejor que "Total" o TOTAL.

- No utilizar espacios ni el guión bajo "_" en el nombre de la variable, es mejor utilizar el punto ".". Así, "suma.total" es un nombre mejor que "suma total" o "suma_total".

- Si es posible, añadir la unidad de medida al nombre de la variable después del nombre. Por ejemplo, "nitrato.ppm".

- Si se introduce una fecha, es mejor escribir el año completo. Así, introducir "01/01/2019" es mejor que introducir "01/01/19".

- No dejar celdas en blanco en nuestra tabla de datos. Rellenar los datos que faltan con "NA".

El objeto Tabla de Datos (Data Frame)

R tiene un objeto que es específico para guardar datos. Se trata de la Tabla de Datos o Data Frame en inglés. Para crear un objeto Tabla de Datos se utiliza la función "data.frame()":

```
> mis.datos <- data.frame()
```

presionamos ENTER para crear el objeto y si inspeccionamos el objeto "mis.datos":

```
> mis.datos
```

Al presionar ENTER, R nos informa de que se trata de un objeto Tabla de Datos vacío:

```
data frame with 0 columns and 0 rows
```

R nos informa que el objeto tiene 0 columnas y 0 filas. Y es que el objeto Tabla de Datos permite agrupar variables de distintos tipos en columnas, asignar nombres a las columnas para que la referencia a las variables sea sencilla y también permite asignar un nombre único a cada fila para distinguir las observaciones.

El editor de datos

El primer método que vamos a aprender para introducir nuestros datos en R es mediante el editor de datos de R. Para crear un objeto Tabla de Datos y editar su contenido utilizaremos la función "edit":

```
> mis.datos <- edit(data.frame())
```

y al presionar ENTER se abrirá el editor de datos en una ventana nueva (Figura 1). El editor de datos se presenta como una tabla con filas y columnas. Si hacemos click con el botón izquierdo del ratón sobre el encabezado de una columna podremos cambiar el nombre y el tipo de variable (Figura 2).

El editor de datos admite variables de dos tipos: “numeric” o numérico para variables cuantitativas y “character” o texto para variables cualitativas. Después de definir el nombre y el tipo de cada variable de nuestra Tabla de Datos, podemos hacer click con el botón izquierdo en las celdas de la tabla para introducir los datos (Figura 3). Podemos introducir los datos de la Tabla 2 para que nos sirva de ejemplo. Al terminar de introducir todos los datos, obtendremos algo así (Figura 4) y podemos cerrar el editor de datos haciendo clic en el botón de cerrar ventana ☒ del editor, en la esquina superior derecha.

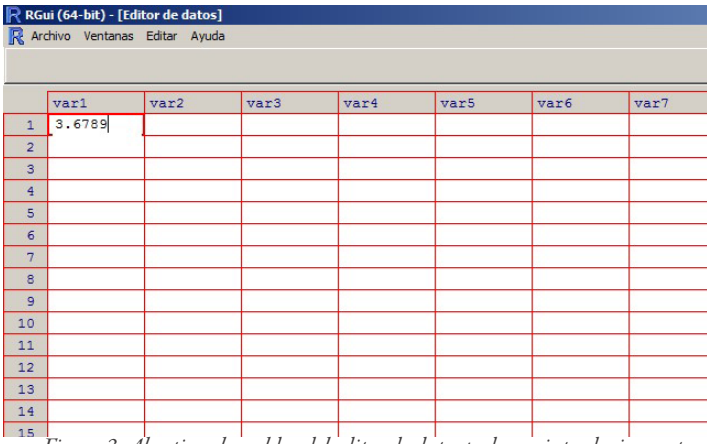


Figura 3. Al activar las celdas del editor de datos podemos introducir nuestros datos en R en columnas de una forma similar a EXCEL.

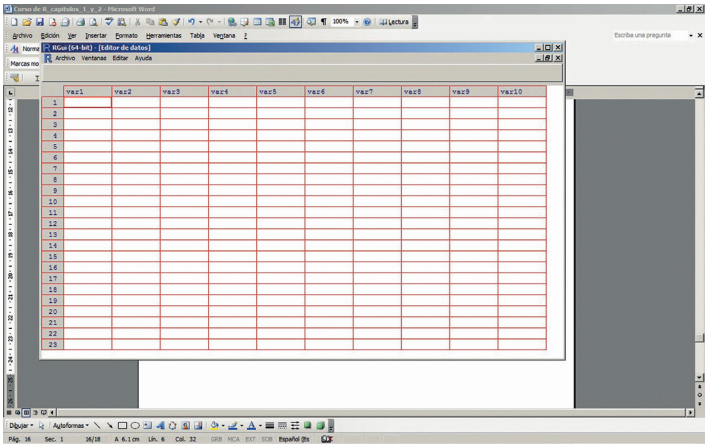


Figura 1. Imagen del editor de datos de R.

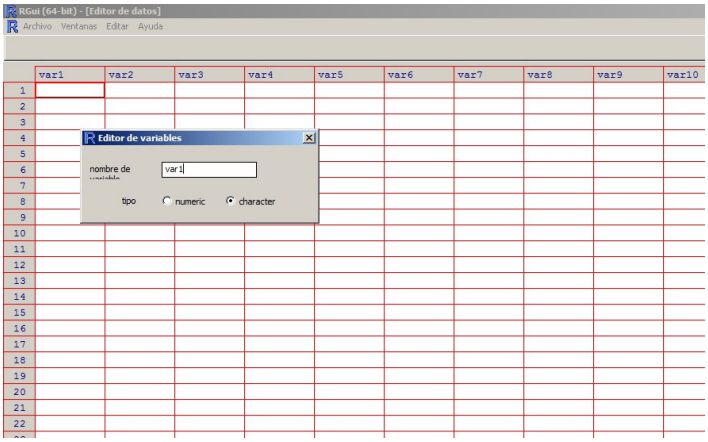


Figura 2. Con el editor de datos podemos cambiar el tipo, numérica o cuantitativa y texto o cualitativa, y el nombre de cada variable.

	planta	grupo	peso.kg
1	p01	ctrl	4.17
2	p02	ctrl	5.58
3	p03	ctrl	5.18
4	p04	ctrl	6.11
5	p05	ctrl	4.5
6	p06	ctrl	4.61
7	p07	ctrl	5.17
8	p08	ctrl	4.53
9	p09	ctrl	5.33
10	p10	ctrl	5.14
11	p11	trat1	4.81
12	p12	trat1	4.17
13	p13	trat1	4.41
14	p14	trat1	3.59
15	p15	trat1	5.87
16	p16	trat1	3.83
17	p17	trat1	6.03
18	p18	trat1	4.89
19	p19	trat1	4.32
20	p20	trat1	4.69
21	p21	trat2	6.31
22	p22	trat2	5.12
23	p23	trat2	5.54
24	p24	trat2	5.5
25	p25	trat2	5.26
26	p26	trat2	5.29
27	p27	trat2	4.92
28	p28	trat2	6.15
29	p29	trat2	5.8
30	p30	trat2	5.26
31			

Figura 4. Datos introducidos en el editor de datos de R

Si volvemos a examinar el objeto "mis.datos":

> mis.datos

al presionar ENTER, R nos muestra el contenido de "mis.datos": los nombres de las columnas o variables, los nombres de las filas u observaciones y los datos contenidos en el objeto. Una forma más conveniente de examinar el contenido de una Hoja de Datos es utilizar la función "str()":

> str(mis.datos)

que al presionar ENTER nos muestra la estructura de la Hoja de Datos:

'data.frame': 30 obs. of 3 variables:

$ planta : chr "p01" "p02" "p03" "p04" ...

$ grupo : chr "ctrl" "ctrl" "ctrl" "ctrl" ...

$ peso.kg: num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...

La función "str()" nos muestra información más completa sobre la Hoja de Datos: el número de observaciones, el número de variables, los nombres de las variables y su tipo. Además, "str()" imprime en la pantalla las primeras líneas del objeto para poder revisar su contenido.

El editor de datos también se puede utilizar para modificar una Tabla de Datos existente. Para modificar el objeto "mis.datos", debemos introducir:

> mis.datos <- edit(mis.datos)

De esta forma, los cambios que realizamos en el editor de datos se van a modificar en el objeto "mis.datos" que ya teníamos creado.

Leer datos desde un fichero con extensión .csv

La función "read.table()":

read.table(file, header=T, sep="", dec=".", row.names)

nos permite leer archivos con la extensión .csv (archivos de valores separados por comas) generados con EXCEL. Los argumentos de la función "read.table()" son (Tabla 3):

- file: argumento de tipo texto, es el nombre del archivo .csv que contiene los datos que queremos importar en R. Para que R pueda localizar este fichero, tiene que estar localizado en el directorio de trabajo. Si no está localizado en el directorio de trabajo, podemos utilizar la función "file.choose()" en lugar del nombre del fichero. Esta función abre el explorador de archivos de Windows y nos permite buscar en archivo en el directorio del ordenador.

- header: argumento de tipo lógico, indica a R si el fichero contiene los nombres de las variables. Si tiene el valor T (verdadero), R utilizará la primera fila del fichero para crear los nombres de las columnas. Si tiene el valor F (falso), R creará nombres de variables por defecto: la letra V seguido del número de columna (por ejemplo: V1, V2, V3, ...).

- sep: argumento de tipo texto, indica a R cuál es el carácter que se utiliza para separar los valores en el fichero. En un fichero de tipo ".csv", los valores de cada variable están separados por el carácter ";". Otros caracteres usuales en ficheros de texto que contienen datos son el espacio o "" y la tabulación o "\t".

Tabla 3. Argumentos de la función "read.table()"

Argumento	Tipo	Ejemplo	Uso
file	texto	file="datos.csv"	El nombre del archivo donde se encuentran los datos que queremos importar a R
header	lógico	header=T	Indica si el fichero incluye el nombre de las variables o no
row.names	numérico	row.names=1	Indica el número de columna que contiene los nombres de las filas
sep	texto	sep=";"	Indica el carácter que se utiliza para separar un dato del siguiente
dec	texto	dec=","	Indica el carácter que se utiliza para separar los decimales de un número

- dec: argumento de tipo texto, indica a R cuál es el carácter que se utiliza para separar los decimales de un número. Puede tener dos valores "." o ",", normalmente va a depender de la configuración del ordenador donde se creó el archivo de texto que contiene los datos. En caso de duda, se puede examinar el archivo ".csv" con un procesador de textos para ver qué carácter se utiliza para separar los decimales.

- row.names: argumento de tipo numérico, indica el número de columna que contiene los nombres de filas y normalmente tendrá el valor 1. Si no se utiliza este argumento, R genera nombres de fila numéricos por defecto (por ejemplo: 1, 2, 3, ...).

Para testar esta función, primero tabulemos los datos de la Tabla 2 en EXCEL y guardemos el resultado como un archivo de tipo .csv con el nombre de "misdatos.csv" (Figura 5). Después podemos utilizar cualquiera de estos dos comandos:

Tabla 2. Ejemplo de datos recogidos en un experimento de laboratorio sobre crecimiento de plantas con un control y dos tratamientos diferentes de fertilizantes. Datos de Dobson (1983).

planta	grupo	peso (kg)	planta	grupo	peso (kg)
p01	cont	4.17	p16	trat1	3.83
p02	cont	5.58	p17	trat1	6.03
p03	cont	5.18	p18	trat1	4.89
p04	cont	6.11	p19	trat1	4.32
p05	cont	4.5	p20	trat1	4.69
p06	cont	4.61	p21	trat2	6.31
p07	cont	5.17	p22	trat2	5.12
p08	cont	4.53	p23	trat2	5.54
p09	cont	5.33	p24	trat2	5.5
p10	cont	5.14	p25	trat2	5.37
p11	trat1	4.81	p26	trat2	5.29
p12	trat1	4.17	p27	trat2	4.92
p13	trat1	4.41	p28	trat2	6.15
p14	trat1	3.59	p29	trat2	5.8
p15	trat1	5.87	p30	trat2	5.26

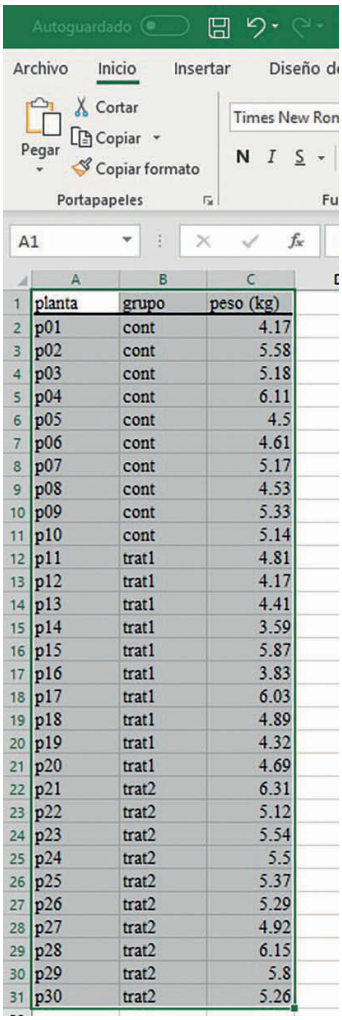


Figura 5. Datos introducidos en EXCEL para crear un fichero .csv. La tabla de datos debe de comenzar en la primera fila y la primera columna de EXCEL, en la celda A1.

```
> mis.datos.2 <- read.table("misdatos.csv", header=T, sep=";",
dec=".", row.names=1)
```

o

```
> mis.datos.2 <- read.table(file.choose(), header=T, sep=";",
dec=".", row.names=1)
```

aunque debemos ajustar el valor del argumento "dec" en función del contenido del fichero. Mi Excel está configurado para utilizar el punto "." y por eso en el ejemplo utilizo sep=".". Si hemos seleccionado bien el archivo, al presionar ENTER se creará un nuevo objeto "mis.datos.2" y podemos examinar su estructura con la función str():

> str(mis.datos.2)

al presionar ENTER podemos ver la estructura de "mis.datos.2":

'data.frame': 30 obs. of 2 variables:

$ grupo : Factor w/ 3 levels "cont","trat1",..: 1 1 1 1 1 1 1 1 1 1 ...

$ peso..kg.: num 4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...

donde podemos ver algunas diferencias con respecto a las estructura de "mis.datos". En primer lugar, sólo tenemos dos variables ¿qué ha sucedido con la variable planta? Podemos examinar el objeto "mis.datos.2":

> mis.datos.2

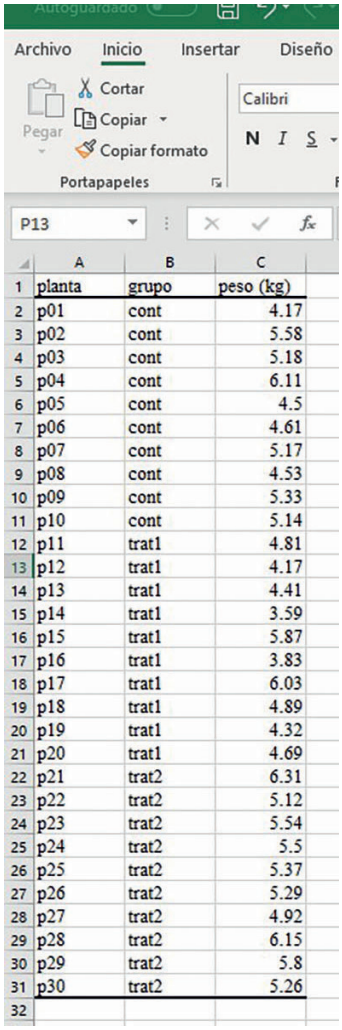
y al presionar ENTER comprobaremos que la variable planta se ha transformado en los nombres de las filas de la Hoja de Datos. Otra diferencia es que la variable grupo no es de tipo texto, sino que es un tipo de dato diferente llamado factor que explicaremos en el próximo capítulo. De momento, sólo necesitamos entender que una variable categórica o cualitativa puede estar representada en R por datos de tipo texto o de tipo factor. Aparte de ficheros de tipo ".csv", con la función "read.table" podemos leer cualquier fichero de texto siempre que ajustemos los valores de los argumentos "sep" y "dec" de forma correcta.

Leer datos directamente desde EXCEL

Si tenemos nuestros datos tabulados en EXCEL, podemos utilizar la función "Copiar" de EXCEL para pegar los datos en R. Primero, debemos seleccionar todos los datos que queremos exportar a R y presionar CTRL+C para copiar los datos al apuntador de Windows (Figura 6). Después debemos utilizar el siguiente comando en R:

> mis.datos.3 <- read.table("clipboard", header=T, sep="\t", dec=".", row.names=1)

pero recordad que tenéis que modificar el argumento "dec" según la configuración del EXCEL que estáis utilizando. Además, se ha sustituido el nombre del fichero por la palabra "clipboard". Al presionar ENTER, se creará el objeto "mis.datos.3" que tendrá la misma estructura que el objeto "mis.datos.2". Aunque esta comprobación ya la podéis hacer vosotros sin ayuda.



	A	B	C
1	planta	grupo	peso (kg)
2	p01	cont	4.17
3	p02	cont	5.58
4	p03	cont	5.18
5	p04	cont	6.11
6	p05	cont	4.5
7	p06	cont	4.61
8	p07	cont	5.17
9	p08	cont	4.53
10	p09	cont	5.33
11	p10	cont	5.14
12	p11	trat1	4.81
13	p12	trat1	4.17
14	p13	trat1	4.41
15	p14	trat1	3.59
16	p15	trat1	5.87
17	p16	trat1	3.83
18	p17	trat1	6.03
19	p18	trat1	4.89
20	p19	trat1	4.32
21	p20	trat1	4.69
22	p21	trat2	6.31
23	p22	trat2	5.12
24	p23	trat2	5.54
25	p24	trat2	5.5
26	p25	trat2	5.37
27	p26	trat2	5.29
28	p27	trat2	4.92
29	p28	trat2	6.15
30	p29	trat2	5.8
31	p30	trat2	5.26
32			

Figura 6. Debemos seleccionar todos los datos que queremos exportar a R y presionar CTRL+C para copiar la tabla de datos al apuntador de Windows.

¿Cómo guardar y recuperar mis datos?

Guardar datos en el directorio de trabajo

Para guardar las Hojas de Datos que acabamos de crear tenemos dos opciones. En primer lugar podemos seleccionar los objetos que queremos guardar con la función "save()":

save(..., file)

que acepta dos argumentos (Tabla 4):

- ...: los puntos suspensivos indican que la función acepta un número variable de argumentos que en este caso son los nombres de los objetos que queremos guardar separados por comas.

- file: argumento de tipo texto, es el nombre de fichero en el que queremos guardar los datos de R. Normalmente, le asignaremos la extensión ".RData" para saber que son datos de R. Podemos utilizar la función "file.choose()" en vez de proporcionar el nombre del fichero.

Tabla 4. Argumentos de la función "save()"

Argumento	Tipo	Ejemplo	Uso
...		objeto.1, objeto.2, ...	Nombres de los objetos que queremos guardar separados por comas
file	Texto	file="datos.RData"	El nombre del archivo donde queremos guardar los datos de R

Para guardar el objeto "mis.datos.2" usaremos:

> save(mis.datos.2,file=file.choose())

al presionar ENTER se abrirá una ventana que nos permite especificar el nombre del archivo y el directorio donde queremos guardarlo. Recordad que es conveniente guardar los ficheros de R con la extensión ".RData", así el nombre del objeto que usaremos para guardar el objeto mis.datos.2 es "misdatos2.RData". Si revisamos la carpeta veremos que se ha creado el archivo.

La segunda opción es guardar el área de trabajo con la función "save.image()" que acepta un argumento "file" como la función "save()":

> save.image(file.choose())

De esta forma guardaremos en un solo fichero todos los objetos que se encuentran en la memoria de R. El resultado de esta función es similar a utilizar el comando "Guardar área de trabajo..." del menú "Archivo" que explicamos en el capítulo I.

Recuperar nuestros datos al inicio de una sesión

Recuperar los datos es sencillo. Tenemos que usar la función "load()" para buscar el archivo que contiene los datos que queremos recuperar. Pata testar esta función, eliminemos primero el objeto "mis.datos.2":

> rm(mis.datos.2)

después de presionar ENTER podemos utilizar la función "ls()" para comprobar que el objeto ya no está almacenado en la memoria de R:

> ls()

Ahora podemos utilizar "load()" para volver a recuperar el objeto "mis.datos.2" en R:

> load(file.choose())

al presionar ENTER se abrirá una ventana para que podamos seleccionar el archivo "misdatos2.RData". Después de cargar el archivo, podemos volver a utilizar la función "ls()" para comprobar que el objeto "mis.datos.2" se encuentra de nuevo en la memoria de R.

En este capítulo hemos aprendido a organizar nuestros datos, introducirlos en R, guardarlos al final de una sesión de trabajo y recuperarlos al inicio de una nueva sesión de trabajo. Con estas habilidades, ya estamos listos para iniciar el análisis de nuestros datos, que será el tema del próximo capítulo.

** Docente - Investigador de la Escuela de Gestión Ambiental, Pontificia Universidad Católica del Ecuador Sede Esmeraldas, Eugenio Espejo y subida a Santa Cruz, 080150 Esmeraldas, Ecuador, contacto: jmolinero2002@yahoo.com*

Referencias

(1) Dobson, A. J. (1983) An Introduction to Statistical Modelling. London: Chapman and Hall.

(2) Greenacre, M. & Primicerio, R. (2014). Multivariate analysis of ecological data. Fundación BBVA, Bilbao.

(3) Owen, W. J. (2010). The R guide. https://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf

(4) Paradis, E. (2003). R para principiantes. https://cran.r-project.org/doc/contrib/rdebuts_es.pdf

(5) Santana, J. S. & Mateos, E. (2014). El arte de programar en R. Un lenguaje para la estadística. https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf

(6) Short, T. (2004). R reference card. https://cran.r-project.org/doc/contrib/Short-refcard.pdf