

Lección 0

Logística de R

R es un entorno de programación para el análisis estadístico y gráfico de datos muy popular, cada día más utilizado en empresas y universidades. Su uso tiene muchas ventajas. Para empezar, es *software* libre. La elección de *software* libre es, en general, acertada por varios motivos: por un lado, transmite valores socialmente positivos, como por ejemplo la libertad individual, el conocimiento compartido, la solidaridad y la cooperación; por otro, nos aproxima al método científico, porque permite el examen y mejora del código desarrollado por otros usuarios y la reproducibilidad de los resultados obtenidos; finalmente, pero no menos importante desde un punto de vista práctico, podemos adquirir de manera legal y gratuita copias del programa, sin necesidad de licencias personales o académicas.

Aparte de su faceta de *software* libre, R tiene algunas ventajas específicas: por ejemplo, su sintaxis básica es sencilla e intuitiva, con la que es muy fácil familiarizarse, lo que se traduce en un aprendizaje rápido y cómodo; además, tiene una enorme comunidad de usuarios, estructurada alrededor de la *Comprehensive R Archive Network*, *CRAN*, que desarrolla cada día nuevos paquetes¹ que extienden sus funcionalidades y cubren casi todas las necesidades computacionales y estadísticas de un científico o ingeniero. En fin, si necesitáis aún más razones para usar R, podéis consultar la página «*Why use R?*» de la organización *inside-R*, en el *url*

<http://www.inside-r.org/why-use-r>

0.1. Cómo instalar R y *RStudio*

Instalar R es muy sencillo; de hecho, seguramente ya lo tenéis instalado en vuestro ordenador, pero es conveniente que dispongáis de su versión más reciente² y que regularmente lo pongáis al día. Los pasos a realizar en Windows o Mac OS X para instalar su última versión son los siguientes:

- (1) Acceded a la página web de la *CRAN*,

<http://cran.r-project.org/>

y pulsad sobre el enlace correspondiente a vuestro sistema operativo: «*Download R for Windows*» o «*Download R for Mac OS X*».

- (2) Si sois usuarios de Windows, entrad en el enlace «*base*», descargad R y seguid las instrucciones de instalación del documento «*Installation and other instructions*» que encontraréis en esa misma página. Si usáis Windows Vista, conviene que consultéis además el enlace «*How do I install R when using Windows Vista?*».

Si sois usuarios de Mac OS X, descargad el fichero `.pkg` correspondiente y, una vez descargado, abridlo y seguid las instrucciones del Asistente de Instalación.

¹ A día de hoy (10/9/2014), el repositorio de paquetes de la CRAN, <http://cran.r-project.org/web/packages/>, contiene 5866 paquetes.

² Las sesiones de R que aparecen en estas notas se han llevado a cabo en la versión 3.1.1.

Si trabajáis con Ubuntu o Debian, para instalar la última versión de R basta que ejecutéis en una terminal, estando conectados a Internet, la siguiente instrucción:

```
sudo aptitude install r-base
```

Cuando instaláis R para Windows o Mac OS X, con él también se os instala una interfaz gráfica que se abrirá al abrir la aplicación y en la que podréis trabajar. La instalación para Linux no lleva una interfaz por defecto, así que sus usuarios tienen que trabajar con R en la terminal (ejecutando R para iniciar una sesión) o instalar aparte una interfaz. Independientemente de todas estas posibilidades, en este curso usaremos *RStudio* como interfaz gráfica de usuario de R para todos los sistemas operativos.

Propiamente hablando, *RStudio* es mucho más que una interfaz de R: se trata de todo un entorno integrado para utilizar y programar con R, que dispone de un conjunto de herramientas que facilitan el trabajo con este lenguaje. Para instalarlo, se ha de descargar de

<http://www.rstudio.com/products/rstudio/download/>

la versión correspondiente al sistema operativo en el que se trabaja. Una vez descargado, en el caso de Windows o Mac OS X ya se puede abrir directamente. En el caso de Linux, hay que ejecutar en una terminal la siguiente instrucción para completar su instalación:

```
sudo dpkg -i rstudio-<version>-i386.deb
```

donde **version** refiere a la versión concreta que se haya descargado.³ Conviene recordar que *RStudio* no es R, ni tan solo lo contiene: hay que instalar ambos programas. De hecho, las instalaciones de R y *RStudio* son independientes una de la otra, de manera que cuando se pone al día uno de estos programas, no se modifica el otro.

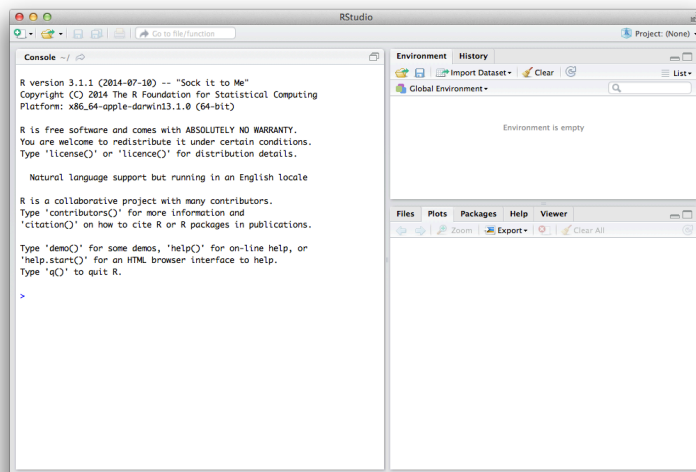


Figura 1. Ventana de *RStudio* para Mac OS X.

Cuando se abre *RStudio*, aparece una ventana similar a la que muestra la Figura 1: su apariencia exacta dependerá del sistema operativo. De momento, nos concentraremos en la ventana de la izquierda, llamada la *consola* de R. Observaréis que en el momento de abrir la aplicación, dicha ventana contiene una serie de información (versión, créditos, etc.) y al final una línea en blanco encabezada por el símbolo >. Este símbolo es la *marca de inicio* e indica

³ A día de hoy (10/9/2014), es la 0.98.1049.

que R espera que escribáis alguna instrucción y la ejecutéis.

Durante la mayor parte de este curso, usaremos *RStudio* de manera interactiva:

- (1) Escribiremos una instrucción en la consola, a la derecha de la marca de inicio de su última línea.
- (2) La ejecutaremos pulsando la tecla *Entrar* (\leftrightarrow).
- (3) R la evaluará y, si corresponde, escribirá el resultado en la línea siguiente de la consola (como veremos, no todas las instrucciones hacen que R escriba algo).
- (4) R abrirá una nueva línea en blanco encabezada por una marca de inicio, donde esperará una nueva instrucción.

Haced una prueba: escribid `1+1` junto a la marca de inicio y pulsad *Entrar*; R escribirá en la línea siguiente el resultado de la suma, `2`, y a continuación una nueva línea en blanco encabezada por la marca de inicio.

```
> 1+1
[1] 2
>
```

Ya hablaremos del `[1]` que precede al `2` en la Lección 3. Hasta entonces, no os preocupéis por él.

Para facilitarnos el trabajo, la consola dispone de un mecanismo para acceder a las instrucciones ya ejecutadas y modificarlas si queremos. Si situamos el cursor a la derecha de la marca de inicio de la línea inferior y pulsamos la tecla de la flecha vertical ascendente \uparrow , iremos obteniendo de manera consecutiva, en esa línea, las instrucciones escritas hasta el momento en la misma sesión; si nos pasamos, podemos usar la tecla \downarrow para retroceder dentro de esta lista; una vez alcanzada la instrucción deseada, podemos volver a ejecutarla o, con las teclas de flechas horizontales, ir al lugar de la instrucción que queramos y reescribir un trozo antes de ejecutarla. Otra posibilidad es usar la pestaña **History** de la ventana superior derecha de *RStudio*, que contiene la lista de todas las instrucciones que se han ejecutado en la sesión actual. Si seleccionamos una instrucción de esta lista y pulsamos el botón «*To console*» del menú superior de la pestaña, la instrucción se copiará en la consola y la podremos modificar o ejecutar directamente.

También podemos copiar instrucciones de otros ficheros y pegarlas a la derecha de la marca de inicio de la manera habitual en el sistema operativo de nuestro ordenador. Pero hay que ir con cuidado: las instrucciones copiadas de ficheros en formato que no sea texto simple pueden contener caracteres invisibles a simple vista que generen errores al intentar ejecutar la instrucción copiada. En particular, esto afecta a las instrucciones que podáis copiar de ficheros en formato PDF como este. Para evitaros este problema, para cada lección encontraréis, en el repositorio del curso, un fichero en formato texto simple llamado `Leccion_i.R` (donde la *i* indica el número de la lección), que contendrá la mayoría de las instrucciones que aparecen en ella. Estas instrucciones las podréis copiar y pegar en la consola, o podréis ejecutarlas directamente abriendo el guión en la ventana de ficheros (véase la Sección 0.3).

Volvamos a la ventana de *RStudio*. Observaréis que está dividida a su vez en tres ventanas. La de la izquierda es la *consola*, donde trabajamos en modo interactivo. La ventana inferior derecha tiene algunas pestañas, entre las que destacamos:

- **Files**, que muestra el contenido de la carpeta de trabajo actual (véase la Sección 0.2). Al hacer clic sobre un fichero en esta lista, se abrirá en la ventana de ficheros (véase la Sección 0.3).
- **Plots**, que muestra los gráficos que hayamos producido durante la sesión. Se puede navegar entre ellos con las flechas de la barra superior de la pestaña.
- **Packages**, que muestra todos los paquetes instalados y, marcados, los que están cargados en la sesión actual (véase la Sección 0.5).
- **Help**, donde aparecerá la ayuda que pidamos (véase la Sección 0.4).

Por lo que refiere a la ventana superior izquierda, tiene dos pestañas:

- **Environment**, con la lista de los objetos actualmente definidos (véase la Lección 1).
- **History**, de la que ya hemos hablado, que contiene la lista de todas las instrucciones que hayamos ejecutado durante la sesión.

Aparte de estas tres ventanas, *RStudio* dispone de una cuarta ventana para ficheros, que se abre en el sector superior izquierdo, sobre la consola (véase la Sección 0.3).

Para cerrar *RStudio*, basta elegir «*Quit RStudio*» del menú «*RStudio*» o pulsar la combinación de teclas usual para cerrar un programa en vuestro sistema operativo.

0.2. Cómo guardar el trabajo realizado

Antes de empezar a utilizar R en serio, lo primero que tenéis que hacer es crear en vuestro ordenador una carpeta específica que será vuestra *carpeta de trabajo* con R. A continuación, en las *Preferencias* de *RStudio*, que podréis abrir desde el menú «*Rstudio*», tenéis que declarar esta carpeta como *Default working directory*. A partir de este momento, por defecto, todo el trabajo que realicéis quedará guardado dentro de esta carpeta, y *RStudio* buscará dentro de esta carpeta todo lo que queráis que lea. Si en un momento determinado queréis cambiar temporalmente de carpeta de trabajo, tenéis dos opciones:

- Podéis usar el menú «*Session → Set Working Directory → Choose Directory...*» para escoger una carpeta.
- Podéis abrir la pestaña **Files** de la ventana inferior derecha y navegar por el árbol de directorios que aparece en su barra superior hasta llegar a la carpeta deseada.

Tanto de una manera como de la otra, la carpeta que especifiquéis será la carpeta de trabajo durante lo que queda de sesión o hasta que la volváis a cambiar.

En cualquier momento podéis guardar la sesión en la que estéis trabajando usando el menú «*Session → Save Workspace as...*». Además, si no habéis modificado esta opción en las *Preferencias*, cuando cerréis *RStudio* se os pedirá si queréis guardar la sesión; si contestáis que sí, *RStudio* guardará en la carpeta de trabajo dos ficheros, *.RData* y *.RHistory*, que se cargarán automáticamente al volver a abrir *RStudio* y estaréis exactamente donde lo habíais dejado.

Nuestro consejo es que digáis que no: normalmente, no os interesará arrastrar todo lo que hayáis hecho en sesiones anteriores. Y si queréis guardar algunas definiciones e instrucciones de una sesión, lo más práctico es guardarlas en un *guión* (véase la Sección 0.3).

Los gráficos que generéis con *RStudio* aparecerán en la ventana inferior derecha, en la pestaña **Plots** que se activa automáticamente cuando se crea alguno. Para guardar un gráfico, hay que ir al menú «*Export*» de esta ventana y seleccionar cómo queréis guardarlo: como una imagen en uno de los formatos estándares de imágenes (.png, .jpeg, .tiff, etc.) o en formato PDF. Entonces, se abrirá una ventana donde podéis darle nombre, modificar sus dimensiones y especificar el directorio donde queráis que se guarde, entre otras opciones.

0.3. Cómo trabajar con guiones y otros ficheros

R admite la posibilidad de crear y usar ficheros de instrucciones que se pueden ejecutar y guardar llamados *guiones* (*scripts*). Estos guiones son una alternativa muy cómoda a las sesiones interactivas, porque permiten guardar las versiones finales de las instrucciones usadas, y no toda la sesión con pruebas, errores y resultados provisionales, y facilitan la ejecución de secuencias de instrucciones en un solo paso. Además, un guión se puede guardar, volver a abrir más adelante, editar, etc. El símbolo # sirve para indicar a R que omita todo lo que hay a su derecha en la misma línea, lo que permite añadir comentarios a un guión.

Para crear un guión con *RStudio*, tenéis que ir al menú «*File* → *New File* → *R Script*». Veréis que os aparece una ventana nueva en la parte superior izquierda de la ventana de *RStudio*, sobre la consola: la llamaremos *ventana de ficheros*. En ella podéis escribir, línea a línea, las instrucciones que queráis. Para ejecutar algunas instrucciones, basta que las seleccionéis y pulséis el botón «*Run*» que aparece en la barra superior de esta ventana.

Para guardar un guión, basta pulsar el botón con el icono de un disquete de ordenador que aparece en la barra superior de su ventana. Otra posibilidad es usar el menú «*File* → *Save*», o pulsar la combinación de teclas usual para guardar un fichero en vuestro sistema operativo, siempre y cuando la *ventana activa* de *RStudio* (donde esté el cursor en ese momento) sea la del guión. Si guardamos un guión concreto por primera vez, se abre una ventana de diálogo donde *RStudio* espera que le demos un nombre; la costumbre es usar para los guiones la extensión .R.

Podéis abrir un guión ya preexistente con *RStudio* usando el menú «*File* → *Open File*» de *RStudio* o pulsando sobre él en la pestaña **Files**. También podéis arrastrar el icono del guión sobre el de *RStudio* o (si habéis declarado que la aplicación por defecto para abrir ficheros con extensión .R sea *RStudio*) simplemente abrir el fichero de la manera usual en vuestro sistema operativo.

Además de guiones, con *RStudio* también podemos crear otros tipos de ficheros que combinen instrucciones de R con instrucciones de otro lenguaje. En este curso lo usaremos para crear ficheros *R Markdown*, que permiten generar de manera muy cómoda informes y presentaciones que incorporen instrucciones de R (o sólo sus resultados). Para crear un fichero *R Markdown*, tenéis que ir al menú «*File* → *New File* → *R Markdown...*», donde os aparecerá una ventana que os pedirá el tipo de documento («*Document*», «*Presentation*...»), su título y el formato de salida. Una vez completada esta información, se abrirá el fichero en la ventana superior izquierda.

Para fijar ideas, supongamos que habéis elegido realizar un informe («*Document*») con formato de salida html; entonces, para generar un informe básico basta sustituir las palabras clave

que ha generado *RStudio* en esta ventana. Probadlo: cambiad el título y el texto; a continuación, guardad el fichero con un nombre y con la extensión `.Rmd`, y pulsad el botón «*Knit html*» situado en la barra superior de la ventana; se generará una ventana aparte con el texto del fichero *R Markdown* en html, que podréis guardar de la manera usual.

Aprender los primeros pasos de *R Markdown* es sencillo. Para ello, id al botón «?» situado en la barra superior de la ventana de ficheros y escoged «*Markdown Quick Reference*»; os aparecerá, en la ventana de ayuda, un manual de referencia rápida de *R Markdown* que se puede leer en 15 minutos y que para la mayoría de ejercicios de este curso es más que suficiente. También os puede ser útil la *chuleta* de *R Markdown* `rm-cheatsheet.pdf` que encontraréis en el repositorio del curso y en su *url* original:

`http://shiny.rstudio.com/images/rm-cheatsheet.pdf.zip`

En cualquier caso, a medida que avance el curso iremos explicando técnicas para mejorar los ficheros resultantes.

0.4. Cómo obtener ayuda

Para conocer toda la información (qué hace, cuál es la sintaxis correcta, qué parámetros tiene, algunos ejemplos de uso...) sobre una función o un objeto, se puede usar la función

`help(nombre del objeto)` o, equivalentemente, `?nombre del objeto`.

Por ejemplo, si entramos⁴ la instrucción

```
> help(sum)
```

obtenemos en la ventana inferior derecha toda la información sobre la función `sum`.

Si queremos pedir información sobre un símbolo, tenemos que escribirlo entrecomillado en el argumento de `help`, o a la derecha de `?`; así, entrando

```
> help("+")
```

obtenemos la información sobre el operador `+`, que, obviamente, sirve para sumar. Si *R* esperaba el nombre del objeto entre comillas y no las incluimos, nos avisará con un mensaje de error.⁵

```
> help(+)  
Error: unexpected ')' in "help(+)"
```

⁴ «Entrar» significa, naturalmente, «escribir a la derecha de la marca de inicio y a continuación pulsar *Entrar*».

⁵ Cuando instaláis *R*, éste reconoce la zona donde vivís a partir de información que extrae del ordenador, y especifica el idioma de los mensajes de acuerdo con esta zona. Seguramente, por lo tanto, los mensajes de error y advertencias os aparecerán en castellano; un castellano algo chapucero, la verdad. Si los queréis en inglés, entrad la instrucción

```
system("defaults write org.R-project.R force.LANG en_US.UTF-8")
```

y, a continuación, cerrad *RStudio* y volvedlo a abrir. Si los tenéis en inglés, o en cualquier otro idioma, y los queréis en castellano, tenéis que usar

```
system("defaults write org.R-project.R force.LANG es_Es.UTF-8").
```

Cuando hayamos avanzado un poco en este curso, el **help** os será muy útil. Aquí sólo veremos alguna aplicación simple de la mayoría de las funciones que estudiemos, con los parámetros más importantes y suficientes para nuestros propósitos, y necesitaréis consultar su **help** para conocer todos sus usos, todos sus parámetros y mucha más información relevante.

Si queremos pedir ayuda sobre un tema concreto, pero no sabemos el nombre exacto de la función, podemos usar

```
help.search("palabra clave") o, equivalentemente, ??palabra clave
```

(las comillas en el **help.search** son obligatorias). Con esta instrucción, conseguiremos en la ventana de ayuda una lista de las funciones que R entiende que están relacionadas con la palabra clave entrada. Entonces, pulsando en la función que nos interese de esta lista, aparecerá la información sobre ella que tendríamos con el **help**. Como podéis imaginar, conviene que la palabra clave esté en inglés.

R dispone también de una página web de ayuda en línea, que se abre entrando la instrucción **help.start()**.⁶

Además de la ayuda que incorpora el mismo R, siempre podéis acudir a foros y listas de discusión para encontrar ayuda sobre cualquier duda que podáis tener. Algunos recursos que nosotros encontramos especialmente útiles son los siguientes:

- La sección dedicada a R del foro **stackoverflow**:

<http://stackoverflow.com/questions/tagged/r>

- El archivo de la lista de discusión *R-help*:

<http://r.789695.n4.nabble.com/r-help-f789696.html>

- El grupo de Facebook «*R project en español*»:

<https://www.facebook.com/groups/rprojectsp>

Es muy probable que alguien ya haya tenido la misma dificultad y se la hayan resuelto en alguno de estos foros.

0.5. Cómo instalar y cargar paquetes

Muchas funciones y tablas de datos útiles no vienen con la instalación básica de R, sino que forman parte de *paquetes* (*packages*), que se tienen que instalar y cargar para poderlos usar. Por citar un par de ejemplos, el paquete **magic** lleva una función **magic** que crea cuadrados mágicos;⁷ para usarla, tenemos que instalar y cargar este paquete. De manera similar, el paquete **ggplot2** incorpora una serie de funciones para dibujar gráficos avanzados que no podemos usar si primero no instalamos y cargamos este paquete.

⁶ Si ejecutáis esta instrucción en otras interfaces de R, esta página de ayuda en línea se abre en un navegador, siempre y cuando estéis conectados a Internet; con *RStudio*, se abre en la pestaña de ayuda y no es necesario estar conectados a Internet.

⁷ Un *cuadrado mágico* es una tabla cuadrada de números naturales tal que las sumas de todas sus columnas, las sumas de todas sus filas y las sumas de sus dos diagonales principales valen todas lo mismo.

Podemos consultar en la pestaña **Packages** la lista de paquetes que tenemos instalados. Los paquetes que aparecen seleccionados en esta lista son los que tenemos cargados en la sesión actual. Si queremos cargar un paquete ya instalado, basta seleccionarlo; podemos hacerlo también desde la consola, con la instrucción

```
library(paquete).
```

En caso de necesitar un paquete que no tengamos instalado, hay que instalarlo antes de poderlo cargar. La mayoría de los paquetes se pueden instalar desde el repositorio del CRAN; esto se puede hacer de dos maneras:

- Desde la consola, entrando la instrucción

```
install.packages("paquete", dep=TRUE)
```

(las comillas son obligatorias). El parámetro `dep=TRUE` obliga a R a instalar no sólo el paquete requerido, sino todos aquellos de los que dependa para funcionar correctamente.

- Pulsando el botón «*Install*» de la barra superior de la pestaña de paquetes; al hacerlo, *RStudio* abre una ventana dónde se nos pide el nombre del paquete a instalar. Conviene dejar marcada la opción «*Install dependencies*», para que se instalen también los paquetes necesarios para su funcionamiento.

Así, supongamos que queremos construir cuadrados mágicos, pero aún no hemos cargado el paquete `magic`.

```
> magic(5)
Error: could not find function "magic"
> install.packages("magic", dep=TRUE) #Instalamos el paquete magic;
    también lo podéis hacer a través de la ventana de paquetes
...
> library(magic) #Cargamos el paquete; también lo podéis hacer a
    través de la ventana de paquetes
> magic(5)
      [,1] [,2] [,3] [,4] [,5]
[1,]    9    2   25   18   11
[2,]    3   21   19   12   10
[3,]   22   20   13    6    4
[4,]   16   14    7    5   23
[5,]   15    8    1   24   17
```

Cuando cerramos *RStudio*, los paquetes cargados en la sesión se pierden; por lo tanto, si queremos volver a usarlos en otra sesión, tendremos que volver a cargarlos.

Hay paquetes que no se encuentran en el CRAN y que, por lo tanto, no se pueden instalar de la forma que hemos visto. Cuando sea necesario, ya explicaremos la manera de instalarlos y cargarlos en cada caso.