



Primeros Pasos con R

José Enrique Martín García

Universidad Politécnica de Gimialcón

(Copyright © 2015)



R lenguaje de programación

Ventajas

- Posibilidad de combinar análisis “empaquetados” en librerías con desarrollos propios
- Gráficos de alta calidad exportables a distintos formatos
- Consume pocos recursos informáticos
- Puede ejecutarse remotamente y conectarse con otros programas
- Foros de programadores y Blogs muy extensa

Desventajas

- Interfaz gráfica limitada
- No hay soporte comercial
- El lenguaje de comandos es un lenguaje de programación



Sesión de R

Una sesión de R se puede iniciar en modo terminal (en sistemas Linux) o empleando un R GUI (en sistemas windows).

La interfaz de R para windows proporciona un menú muy básico para gestionar algunos aspectos:

- El histórico de comandos (.Rhistory)
- El espacio de trabajo de la sesión (.RData)
- La instalación de librerías desde un repositorio CRAN



Hacemos doble clic sobre el ícono de R que aparece en el escritorio



El ambiente de trabajo en R

Al abrir R se mostrará la siguiente imagen

```
RGui (64-bit)
Archivo  Editar  Visualizar  Misc  Paquetes  Ventanas  Ayuda

R Console

R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

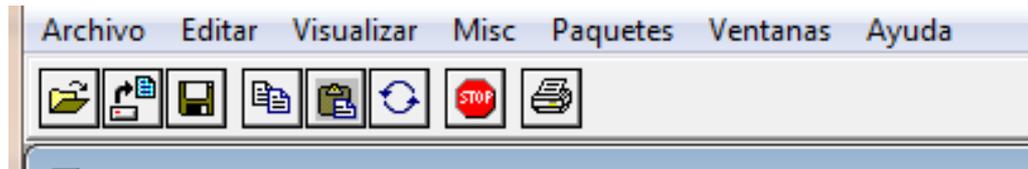
R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

[Previously saved workspace restored]

> |
```



En la imagen podemos identificar los siguientes elementos:

- **El menú principal**

Compuesto por los menús: Archivo, Editar, Visualizar, Misc, Paquetes, Ventanas, y Ayuda. Al desplegar estos menús, podemos realizar procedimientos complementarios a la escritura de programas en R.

- **Los íconos de funciones**

Constituyen accesos abreviados o rápidos a las funciones más usadas de R, como: abrir archivos de programas (documentos con extensión *.txt, *.R); cargar espacios de trabajo (archivos con extensión *.RData); copiar; pegar; copiar imprimir.

-La consola

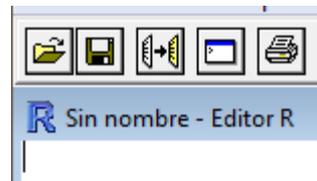
La consola es el espacio en donde: en letras rojas, aparecen las instrucciones dadas a R y en letras azules, sus resultados.

Otros elementos de R, son visibles al realizar procedimientos previos:

-La ventana Script

Para obtener esta ventana, desplegamos el menú Fichero ->Nuevo Script.

Como podemos observar, al activar la ventana Script, los menús Fichero y Editar muestran opciones sólo de esta ventana. También se reduce el número de íconos disponibles y se presentan los íconos: Ejecutar, y cambiar a Consola





La Ayuda en R

R dispone de una ayuda muy completa sobre todas las funciones, procedimientos y elementos del lenguaje:

- Tecllea `? help`
- Tecllea `?? median` o `help.search("median")` para cualquier asunto
- `help(command)` o `?command` si conoces el nombre del comando
- `? '&&'` (o `? "&&"`) para operadores o palabras como `if`
- `help.start()` para obtener documentación de R online
- Cuando no sepamos el nombre exacto del comando o de la función que necesitamos, pero sí que sepamos el tema sobre el que queremos ayuda (en inglés), podemos utilizar la función `help.search()`.
- El menu de ayuda de windows te da multitud de opciones
- El comando `demo()` permite realizar demostraciones del uso de R.



Salvar e iniciar sesiones anteriores

- Se puede guardar una imagen del espacio de trabajo realizado hasta ese momento Desde el [menú Archivo>guardar área de trabajo](#). La extensión para estos archivos es **.RData**.
- Para salir de R podemos seleccionar [Salir del menú Archivo](#), ejecutar la orden `q()` o cerrar directamente la ventana del programa (o la consola). En ese momento, R siempre pregunta si el usuario quiere guardar el actual espacio de trabajo (si se acaba de hacer ya no es necesario).
- Se puede retomar el espacio de trabajo de otra sesión de trabajo abriendo desde el [menú Archivo>cargar área de trabajo](#) la imagen anteriormente guardada, para restaurar los objetos creados en otras sesiones; también es posible haciendo doble click sobre el fichero **.RData**.
- Es posible guardar todos los comandos de la sesión desde el [menú Archivo>Guardar Histórico](#). La extensión de estos ficheros es **.RHistory**.

Salvar e iniciar sesiones anteriores

- Para salvar objetos del espacio de trabajo **x** e **y**, en un fichero llamado `xy.RData` techea `save(x, y, file = "xy.RData")`
- Para salvar el espacio de trabajo, techea `save.image()` o utiliza las opciones de menu **Archivo/Guardar Area de Trabajo**. R proporcinará un nombra de fichero.
- Para cargar el fichero que has salvado previamente "`xy.RData`", techea `load("xy.RData")`
- Una buena práctica es salvar los trabajos de vez en cuando, cuando estemos trabajando con modelos complejos que lleven mucho tiempo en su ejecución. Tambien se podrían guardar datos concretos en un fichero para después reutilizarlos.



Espacio y directorio de trabajo

- El “workspace” es el espacio de trabajo en que se incluyen todos los objetos definidos por el usuario y se almacena en memoria intermedia mientras trabajas con R.
- Cuando termina una sesión de R el propio R te pregunta si quieres guardar el “workspace” para usos futuros. Este espacio, “workspace”, se recarga al volver a iniciar la sesión.
- Directorio de trabajo o “working directory” es el directorio donde por defecto “lee” R. También es donde guardará el workspace al finalizar la sesión y donde buscará un workspace guardado al inicio. Si quieres que R lea un fichero que no esté en “working directory” hay que especificar la ruta completa.

Espacio y directorio de trabajo

• Funciones para manejar el “workspace”:

<code>getwd()</code>	Muestra el directorio de trabajo wd
<code>setwd(“midirectorio”)</code>	Ajusta el wd al especificado
<code>ls()</code> o <code>dir()</code>	Lista lo que hay en el wd
<code>history()</code>	Muestra los últimos comandos ejecutados
<code>savehistory()</code>	Guarda el historial de comandos, por defecto en <code>.RHistory</code>
<code>loadhistory()</code>	Carga el historial de comandos <code>save.image(“myworkspace.R”)</code>
	Guarda los objetos del workspace, por defecto en <code>.RData</code>
<code>load(“myworkspace.R”)</code>	Carga el workspace <code>myworkspace.R</code>



Pasos Básicos con R

- Abriendo R se abre la consola o ventana de comandos de R en la cual introduciremos los comandos y será dónde se verán los resultados de los análisis.
- Aparece una línea en blanco con el símbolo > en el margen izquierdo: PROMPT.
- A partir de aquí R espera que escribamos COMANDOS e instrucciones para comenzar a trabajar.
- Para ejecutar un comando escrito utilizamos la tecla de intro.
- El signo # indica la introducción de un comentario.
- Las órdenes elementales consisten en expresiones o en asignaciones:
 - Si una orden consiste en una **expresión**, se evalúa, se imprime y su valor se pierde.
 - Una **asignación** evalúa una expresión, no la imprime y guarda su valor en una variable.



Pasos Básicos con R

- En cada línea sólo caben 128 caracteres: si queremos escribir más, podemos utilizar otra línea que comienza con el símbolo `+` en el lado izquierdo de la pantalla en vez de `>`.
- Si encontramos un error y nos dice la falta de algo podemos teclearlo, y al pulsar **intro**, ejecutará lo que hayamos solicitado.
- Si lo que ocurre es que nos hemos equivocado, al pulsar la tecla **Esc** el prompt `>` volverá a aparecer.
- Con la tecla `↑` recuperamos las instrucciones utilizadas en la sesión y con las teclas `←→` podemos corregir errores.
- Para separar expresiones se emplea `;` (punto y coma).
- La combinación **Ctrl-C** interrumpe la edición o ejecución en curso.
- **q()** es el comando para salir de R.



Pasos Básicos con R

Desde R se pueden realizar sencillas operaciones aritméticas:

`(2 - 5) * 3/5`

`sqrt (9)`

`sin (pi/2)`

`factorial (3)`

R ejecuta ciertas operaciones que, en otros programas, podrán generar mensajes de error:

`1/0`

`0/0`

`log (-1)`

`sqrt (-9)`



Pasos Básicos con R

En R empleamos el operador '`<-`' para hacer asignaciones, y una variable se crea "al vuelo", esto es, en el mismo instante en el que la asignas.

```
a <- log (10)
```

```
a <- log (10)
```

```
a = log (10) # cual usar := o <-?
```

```
a <- rnorm (4, mean = 10, sd = 1)
```

en R se utiliza = para asignar valores a los argumentos de las funciones

Todo lo precedido por almohadillas '#' R lo considera un comentario y no lo "interpreta".

Al introducir los comandos anteriores, hemos creado un objeto en R: la variable "a".

Este objeto se almacena en una zona de memoria llamada "espacio de trabajo".

```
ls ()
```



R como calculadora

Podemos emplear R como una calculadora.

```
45 + 23
```

```
[1] 68
```

```
sqrt(25)
```

```
[1] 5
```

```
# usando variables
```

```
x <- 100/4
```

```
[1] 5
```

```
pi
```

```
[1] 3.142
```

```
r = 5
```

```
area <- 2 * pi * r
```

```
area
```

```
[1] 11.28
```

```
c(25, 100, 2, 3) * 5
```

```
[1] 125 500 10 15
```

R: Trabajos con scripts

Para cargar un fichero con instrucciones, un script, empleamos la instrucción/función `'source()'`

```
source("fichero_de_comandos.r")
```

Un fichero que crease un histograma de una curva normal debería de contener, por ejemplo, estas instrucciones:

```
x <- rnorm(500)
# crea 500 observaciones aleatorias de una normal tipificada
hist(x)
# hace la gráfica de la distribución
```

Si creamos un fichero de texto plano con el contenido anterior, lo guardamos con el nombre "distrib_normal_500.R".

Ejecutamos:

```
source("distrib_normal_500.R")
```