

# CURSO DE R (I) CONCEPTOS BÁSICOS DE R

Por Jon Molinero Ortiz \*\*



## ¿Qué es R?

R es un programa gratuito para realizar análisis y gráficos estadísticos. Se trata de un lenguaje de programación con funciones para realizar análisis estadísticos básicos y avanzados. Puede utilizarse con Windows, pero está disponible también para otros sistemas operativos como Linux o Mac OS. R es muy utilizado por investigadores de todo el mundo debido a la fiabilidad de sus métodos de cálculo y a que se puede descargar de forma libre.

## 4 razones para utilizar R

1) R se distribuye bajo una licencia de software libre y es completamente gratuito. El usuario tiene a su disposición una amplia gama de métodos estadísticos y gráficos sencillos y avanzados a un coste cero.

2) Ha sido creado por matemáticos para matemáticos. Los métodos de cálculo que utiliza son de gran robustez, han sido profesionalmente testados y son revisados y ampliados continuamente.

3) R se puede personalizar fácilmente mediante paquetes descargables que contienen funciones adicionales. Un investigador de ciencias sociales y otro de ciencias experimentales pueden ambos utilizar R, pero seguramente utilizarán paquetes de funciones completamente diferentes.

4) Existe una gran comunidad de usuarios de R y se pueden encontrar con facilidad foros y páginas sobre funciones específicas de R en línea.

## Descargar e instalar R

La página de R es: <https://www.r-project.org/> (sólo en inglés, lo siento). Desde aquí, buscaremos el link para descargar R “download R”. Primero, debemos elegir el servidor que vamos a utilizar, normalmente un servidor que se encuentre geográficamente cerca de nuestro lugar de trabajo.

Tras seleccionar el servidor, nos aparece una ventana donde podemos seleccionar la versión de R adecuada a nuestro sistema operativo:

“Download R for Linux”  
“Download R for (Mac) OS”  
“Download R for Windows”

Al seleccionar el sistema operativo Windows, se nos abrirá una nueva ventana y debemos seleccionar “install R for the first time”. Se nos abrirá una nueva ventana donde aparece el enlace para descargar la versión estable más actual de R “Download R X.X.X for Windows”. Haciendo doble click con el ratón en el fichero que hemos descargado, podremos instalar R.

Durante el proceso de instalación, podemos elegir el idioma de trabajo del programa. Aunque prácticamente la totalidad de la información sobre R se encuentra en inglés, podemos instalar una versión de R en español.

## Inicio de sesión de trabajo

Para iniciar R simplemente hacemos doble click en el icono del programa que hay en el escritorio (es indiferente seleccionar la versión i386 o x64) y se iniciará el programa (Figura 1). Al iniciar R, la ventana de trabajo se muestra normalmente disminuida. Para facilitar el trabajo, es conveniente hacer click en el icono de maximizar ventana (□).

Una buena costumbre de uso cuando se trabaja con R es elegir la carpeta de trabajo al inicio de la sesión. Para elegir la carpeta de trabajo utilizaremos el menú “Cambiar dir ...”:

## Archivo >> Cambiar dir ...

Aunque cada uno podéis organizar el trabajo de R como queráis, resulta más eficiente guardar cada sesión de trabajo en la carpeta del proyecto correspondiente que establecer una carpeta que se llame “Sesiones R”, y guardar en esa carpeta la información de diferentes sesiones de trabajo.

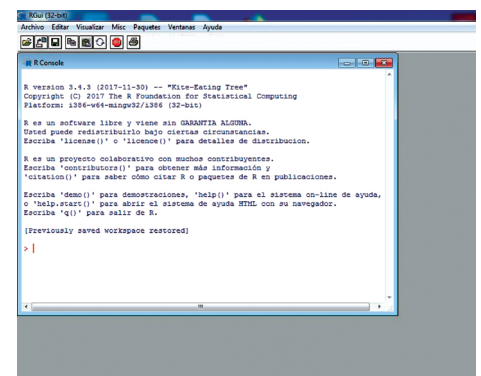


Figura 1. Pantalla de inicio de R

## ¿Cómo se utiliza R?

### El editor de comandos

La mayor parte del uso de R no se realiza a través de menús desplegables como en las aplicaciones comunes de Windows y lo primero que nos llama la atención cuando iniciamos el programa es la escasez de menús y de botones en la ventana de trabajo. En R, las instrucciones se introducen con el teclado en un editor de comandos que aparece en la ventana de trabajo debajo de varias líneas de texto con información general. Cuando el editor de comandos está listo para aceptar nuestras instrucciones se muestra como el símbolo



“>” y un cursor parpadeante:

```
> |
```

El editor de comandos permite introducir instrucciones con el teclado y al presionar la tecla ENTER, el programa realiza los cálculos que hemos solicitado. Por ejemplo, si introducimos una operación de suma sencilla:

```
> 2+5
```

y presionamos ENTER, R nos muestra el resultado de la operación:

```
[1] 7
```

R reconoce los operadores matemáticos comunes y de esta forma, podemos utilizar R para realizar cálculos sencillos (Tabla 1). Para realizar operaciones matemáticas más complejas y análisis estadísticos debemos aprender a utilizar las funciones de R.

Tabla 1. Operadores matemáticos en R

Función de R	Uso	Ejemplo	Resultado
+	Suma	> 3+2	5
-	Resta	> 3-2	1
*	Multiplicación	> 3*2	6
/	División	> 3/2	1.5
^	Exponente	> 3^2	9
()	Paréntesis	> (3+2)*2	10

## Funciones

Introducir funciones en el editor de comandos es la forma de dar órdenes a R para realizar cálculos matemáticos más complejos, crear gráficas y hacer análisis estadísticos. Una función de R tiene siempre esta estructura:

*nombre\_de\_la\_función* (*argumento\_1*, *argumento\_2*, *argumento\_3*, ...)

El nombre de la función y los paréntesis siempre se utilizan cuando introducimos funciones en el editor de comandos. La información adicional que la función necesita son los argumentos. El número de argumentos es variable y hay funciones sin argumentos y otras con uno, dos, tres o más argumentos. Cuando la función necesita 2 o más argumentos, entonces éstos se separan con una coma “,”.

Por ejemplo, para obtener el logaritmo decimal de un número se utiliza la función:

```
> log10(100)
```

El nombre de la función es “log10” y sólo utiliza un argumento, en este caso “100”. Si presionamos ENTER obtenemos el cálculo indicado:

```
[1] 2
```

Utilicemos ahora una función con dos argumentos:

```
> log(100, 10)
```

Esta función tiene dos argumentos. El primer argumento “100” es el número del que queremos conocer su logaritmo y el segundo argumento “10” es la base sobre la cual se calcula el logaritmo.

Si presionamos ENTER obtenemos:

```
[1] 2
```

Las funciones pueden ser a su vez argumentos de otras funciones, por ejemplo:

```
> exp(log(10))
```

Al presionar ENTER obtendremos:

```
[1] 10
```

¿Qué ha sucedido? Primero, la función “log” ha calculado el logaritmo neperiano de 10 y después, la función “exp” ha utilizado el resultado de la función “log” como exponente y ha elevado  $e$  a ese número. El resultado es 10.

Tabla 2. Funciones matemáticas comunes de R

Función de R	Uso	Ejemplo	Resultado
$\log(x)$	Logaritmo neperiano de x	> log(10)	2.3025
$\log(x,n)$	Logaritmo de x en base n	> log(10, 2)	3.3219
$\log_{10}(x)$	Logaritmo de x en base 10	> log10(10)	1
$\text{sqrt}(x)$	Raíz cuadrada de x	> sqrt(25)	5
$\text{factorial}(x)$	Calcula x!	> factorial(4)	24
$\cos(x)$	Coseno del ángulo x en radianes	> cos(pi/4)	0.7071
$\sin(x)$	Seno del ángulo x en radianes	> sin(pi/4)	0.7071
$\tan(x)$	Tangente del ángulo x en radianes	> tan(pi/4)	1
$\text{acos}(x)$	Ángulo en radianes cuyo coseno es x	> acos(1)	0
$\text{asin}(x)$	Ángulo en radianes cuyo seno es x	> asin(1)	1.5708
$\text{atan}(x)$	Ángulo en radianes cuya tangente es x	> atan(1)	0.7854
$\text{abs}(x)$	Valor absoluto de x	> abs(-10)	10



R tiene funciones matemáticas similares a las que encontramos en una calculadora (Tabla 2) que se pueden combinar con los operadores matemáticos para generar expresiones complejas:

```
> 4+log(2.34)-sin(2+3)
```

## Objetos

Hasta ahora hemos aprendido a utilizar R como una calculadora. Hemos realizado cálculos sencillos, pero no podemos guardarlos de forma permanente. Deberíamos ser capaces de guardar la información que genera R de forma permanente para que la podamos recuperar y reutilizar en diferentes sesiones de trabajo.

R utiliza objetos para guardar la información en la memoria del ordenador. Un objeto puede contener un número, una tabla de datos e incluso un análisis estadístico completo.

Para crear un objeto de R, primero debemos pensar un nombre y utilizar el operador de asignación “<-“ para indicarle al editor qué queremos guardar en el objeto.

Creemos un objeto sencillo:

```
> a <- 2
```

al presionar ENTER no obtenemos ninguna respuesta de R porque el valor “2” se ha almacenado en la memoria del ordenador. Para ver el contenido de un objeto, sólo tenemos que teclear el nombre del objeto en el editor:

```
> a
```

al presionar ENTER, el editor nos devuelve la información que contiene

el objeto a que está almacenado en la memoria del ordenador:

```
[1] 2
```

Si intentamos obtener información de un objeto que no existe:

```
> b
```

al presionar ENTER, R nos devuelve un mensaje de error:

```
Error: objeto 'b' no encontrado
```

Para nombrar los objetos podemos utilizar cualquier nombre que se nos ocurra, pero es una costumbre de buen uso utilizar nombres en minúscula y si se componen de varias palabras, separar éstas con un punto “.”. R distingue entre letras minúsculas y mayúsculas y por lo tanto los nombres siguientes son correctos, pero hacen referencia a objetos diferentes, aunque el nombre sea idéntico:

```
mi.objeto  
Mi.objeto  
MI.OBJETO
```

También es importante que el nombre del objeto nos dé información de su contenido. Los nombres siguientes son correctos:

```
tabla.datos  
mis.datos  
resultados
```

pero es más conveniente si el nombre nos informa del contenido:

```
tabla.macroinvertebrados  
datos.química  
correlación.química
```

Los objetos pueden almacenar el resultado de expresiones matemáticas y funciones. Estos objetos son correctos:

```
> b <- 4*(5+6)  
> c <- log10(100)
```

Los objetos también pueden ser argumentos para las funciones, por ejemplo:

```
> log10(b)
```

Finalmente, podemos asignar el contenido de un objeto a otro objeto, e incluso, un objeto puede asignarse a sí mismo.

Todas estas creaciones de objetos son correctas:

```
> d <- 10  
> e <- d  
> e <- e+10
```

Como resumen de esta sección, creemos un objeto para almacenar nuestra edad:

```
> mi.edad <- 48
```

Podemos revisar cómo se ha almacenado este valor en la memoria del ordenador:

```
> mi.edad
```

y presionando ENTER obtenemos lo siguiente:

```
[1] 48
```

Ahora supongamos que llega nuestro cumpleaños y tenemos que manipular el objeto “mi.edad”:

```
> mi.edad <- mi.edad+1
```



Al presionar ENTER no obtendremos ninguna respuesta, pero si volvemos a revisar el valor de “mi.edad”:

```
> mi.edad
```

al presionar ENTER obtendremos lo siguiente:

```
[1] 49
```

hemos sido capaces de modificar el valor almacenado en la memoria del ordenador. A continuación, utilicemos la función “ls”. Se trata de una función que no tiene argumentos que sirve para obtener un listado de todos los objetos que hay en la memoria y están siendo utilizados por R:

```
> ls()
```

al presionar ENTER obtendremos un listado de objetos entre los que se encontrará “mi.edad”.

Ahora podemos utilizar el comando “remove” o su abreviatura “rm” que sirve para borrar objetos:

```
> rm(mi.edad)
```

Si volvemos a utilizar el comando “ls” podremos ver que el objeto “mi.edad” ha sido eliminado. Ya no se encuentra disponible en R. Ya hemos aprendido a crear, a examinar y a manipular los objetos de R y a obtener un listado de los objetos que se encuentran en memoria, y a borrar los objetos que no nos interesan.

## Navegación

El editor de R tiene un sistema de navegación que nos permite revisar todas las funciones que hemos utilizado durante una sesión de trabajo utilizando las teclas del cursor  $\uparrow$  y  $\downarrow$ . Una vez que hemos

seleccionado la función que queremos reutilizar podemos utilizar las teclas del cursor  $\leftarrow$  y  $\rightarrow$  para mover el cursor dentro de la función y modificarla como deseemos. Al presionar ENTER, el editor realiza la función que hemos seleccionado. El editor de comandos tiene dos modos de funcionamiento: línea sencilla o multilínea. En el modo multilínea podemos introducir instrucciones que ocupan más de una línea de texto del editor de comandos, pero este modo apenas se utiliza y normalmente utilizaremos comandos de una sola línea. Cuando se activa el uso de instrucciones multilínea, el cursor del editor de comandos cambia a:

+

Si se activa el modo multilínea del editor de comandos por error, debemos pulsar la tecla ESCAPE y a continuación la tecla ENTER para que el editor vuelva al modo de línea sencilla.

## Personalizar R

R se puede personalizar a través de paquetes gratuitos que contienen funciones que extienden las capacidades básicas de R. Para poder utilizar estos paquetes, primero se deben instalar en la librería interna de R. Para ello, tenemos que utilizar el menú “Instalar paquetes(s)...”:

### Paquetes >> Instalar paquetes(s) ...

Se abrirá inicialmente una ventana para que seleccionemos el servidor de R que queremos utilizar, normalmente un servidor que se encuentre geográficamente cerca del lugar donde nos encontramos o un servidor en Estados Unidos o Reino Unido que siempre están operativos.

Para instalar un paquete, elijamos un servidor localizado en Cambridge en el Reino Unido (Figura 2). Al hacer click en “Aceptar”, aparece una nueva ventana donde aparecen todos los paquetes disponibles para R en orden alfabético, donde podemos seleccionar el paquete que nos interesa o presionando la tecla de control, nos permite seleccionar varios paquetes de forma simultánea. De momento, seleccionemos el paquete “vegan” que contiene funciones útiles para el análisis de comunidades biológicas y sus relaciones con variables ambientales (Figura 3). Al hacer click en “Aceptar”, el paquete se descarga y se descomprime automáticamente.

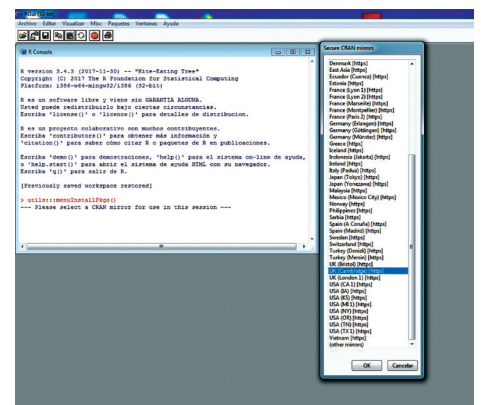


Figura 2. Ventana para seleccionar el servidor para la descarga de paquetes de R

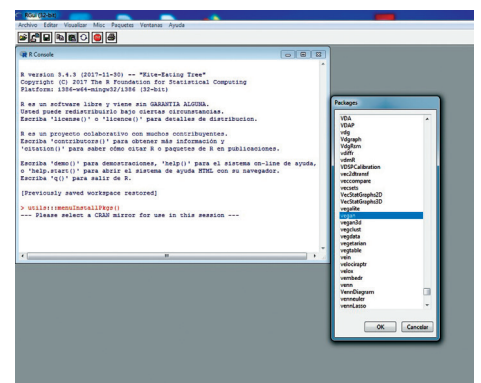


Figura 3. Ventana para la selección de paquetes de R

De esta forma, el paquete queda almacenado en la librería de R y podemos cargarlo en el programa cuando lo necesitemos.



Para cargar un paquete ya instalado desde la librería utilizamos la función “library”:

```
library(nombre_del_paquete)
```

que tiene un argumento, el nombre del paquete que queremos utilizar.

Por ejemplo, podemos cargar el paquete “vegan” que acabamos de instalar:

```
> library(vegan)
```

y R nos proporciona información sobre el paquete que se carga en la memoria, como otros paquetes necesarios para el funcionamiento de “vegan” y que se cargan de forma automática y la versión del paquete:

```
Loading required package:
permute
```

```
Loading required package:
lattice
```

```
This is vegan 2.4-6
```

Las funciones del paquete “vegan” estarían listas para que podamos utilizarlas a través del editor de comandos.

Otra forma de cargar paquetes en R es utilizar el menú “Cargar paquete...”:

### Paquetes >> Cargar paquete ...

Se abrirá una ventana con todos los paquetes que se encuentran en la librería de R y podemos seleccionar el paquete o paquetes que nos interesan. Al hacer click en “Aceptar”, el resultado será el mismo que al utilizar la función “library”.

### Obtener ayuda

¿Dónde puedo encontrar más infor-

mación sobre las funciones de R? Seguro que el programa R os ha cautivado y muchos os estáis haciendo esta pregunta. Afortunadamente, los creadores de R incorporaron un menú “Ayuda” en el programa. Dentro del menú de ayuda, las opciones más útiles quizás sean los manuales en PDF:

### Ayuda >> Manuales (en PDF)

Entre éstos, se encuentra un manual de introducción a R (en inglés) que resulta muy útil para aprender el uso básico del programa. La ventaja de los manuales en PDF es que se pueden descargar y consultar sin tener abierto R. Otro recurso muy útil es la ayuda en html:

### Ayuda >> Ayuda html

Al utilizar este menú, se abrirá el navegador que tengamos configurado por defecto en nuestro ordenador y aparecerá una página con enlaces para acceder a todos los manuales de R. La ventaja de esta ayuda html es que se puede navegar fácilmente entre los diferentes documentos de ayuda de R.

También existe la posibilidad de utilizar la ayuda html desde el editor de comandos usando los símbolos “?” y “??” de esta forma:

```
?nombre_de_función o ??palabra_clave
```

En el primer caso, se nos abrirá la ayuda html de la función. Por ejemplo:

```
> ?aov
```

Nos mostrará la ayuda de la función aov que sirve para hacer análisis de la varianza en R. Si la función que buscamos no existe en R o no existe en los paquetes

que tenemos cargados en la sesión actual, el editor de comandos nos devolverá un error. Cuando utilizamos “??”:

```
> ??aov
```

R buscará en todos los documentos de ayuda y se nos abrirá una página del navegador que muestra todas las páginas de ayuda que tienen alguna referencia a “aov”. En este caso, “aov” no se interpreta como un nombre de función, sino como una palabra clave que se busca en todos los documentos de ayuda de R.

Ya hemos explicado que R se puede personalizar mediante paquetes. Cada uno de estos paquetes tiene uno o varios manuales de ayuda en formato PDF que normalmente explican las funciones que contiene el paquete con numerosos ejemplos de uso.

Para acceder a estos manuales se usa la función “vignette”. Si utilizamos la función “vignette” sin argumentos:

```
> vignette()
```

se abrirá una ventana en R con una lista de todos los manuales que están disponibles en la sesión de trabajo actual.

Para visualizar estos manuales usamos la misma función, pero tenemos que añadir el nombre del manual que queremos abrir como argumento:

```
vignette("nombre_del_manual")
```

Por ejemplo, para abrir el manual de introducción al paquete “vegan”, utilizaremos:

```
> vignette("intro-vegan")
```



Finalmente, una de las formas más fáciles de buscar ayuda sobre R es realizar la búsqueda en GOOGLE. Existe una gran cantidad de información sobre R con muchos ejemplos de uso que ha sido creado por los propios usuarios de R.

También, he creado un grupo de Facebook llamado “useR PUCESE” para que sirva de lugar de encuentro y discusión de los usuarios de R en la PUCESE. La dirección de este grupo es: <https://www.facebook.com/groups/988468334552316/>.

### Fin de la sesión de trabajo

Ya hemos terminado nuestra sesión de trabajo de hoy, pero ¿cómo podemos guardar nuestros objetos para poder seguir utilizándolos en otras sesiones? R permite guardar nuestros objetos y los diferentes comandos que hemos utilizado en una sesión.

Para guardar los objetos que hemos creado durante una sesión de trabajo debemos utilizar el menú “Guardar área de trabajo...”:

#### Archivo >> Guardar área de trabajo

...

Se abrirá un cuadro de diálogo y podremos guardar todos los objetos de R en un fichero con la extensión “.RData”. Para guardar las funciones que hemos utilizado en R durante la sesión usaremos el menú “Guardar histórico...”:

#### Archivo >> Guardar histórico ...

Se abrirá un cuadro de diálogo donde podemos guardar las funciones en un fichero con la extensión “.RHistory”.

Finalmente, podemos cerrar R como se cierra cualquier programa de Windows o, como ya nos hemos acostumbrado al uso del editor de comandos, podemos teclear la función “q”:

```
> q()
```

que al presionar ENTER cerrará el programa. Al cerrar R, siempre nos aparece un cuadro de diálogo que nos pregunta si queremos guardar el área de trabajo.

Normalmente responderemos que no, porque una buena costumbre de uso en R es guardar siempre las sesiones de trabajo en ficheros .RData y .RHistory.

Con esto terminamos de repasar los conceptos básicos de R. En el próximo capítulo aprenderemos a introducir o importar y guardar nuestros datos. 🍀

---

**\*\* Docente investigador de la Escuela de Gestión Ambiental de la PUCE-SE**

### Referencias

1. Owen, W.J. *The R guide*. 2010. Recuperado de: <https://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf>
2. Paradis, E. *R para principiantes*. 2003. Recuperado de: [https://cran.r-project.org/doc/contrib/rdebuts\\_es.pdf](https://cran.r-project.org/doc/contrib/rdebuts_es.pdf)
3. Santana, J. S. & Mateos, E. *El arte de programar en R. Un lenguaje para la estadística*. 2014. Recuperado de: [https://cran.r-project.org/doc/contrib/Santana\\_El\\_arte\\_de\\_programar\\_en\\_R.pdf](https://cran.r-project.org/doc/contrib/Santana_El_arte_de_programar_en_R.pdf)
4. Short, T. *Reference card*. 2004. Recuperado de: <https://cran.r-project.org/doc/contrib/Short-refcard.pdf>