

# MANEJO BÁSICO DE R

Seguro que has empezado las prácticas de laboratorio y has tenido algunos problemas para manejar un programa que no has usado nunca. En este documento te enseñaremos algunas funciones básicas para que sepas usar R perfectamente.

1º Es indiferente poner un símbolo **igual** (=) que poner una **flecha** (←)

```
ii = 3+5
ii
ff <- 3+5
ff
```

Script

```
> ii = 3+5
> ii
[1] 8
> ff <- 3+5
> ff
[1] 8
```

Consola

2º Para poder poner **comentarios** y así aclararnos de qué estamos haciendo, podemos usar las almohadillas (#). Son comentarios que no se tienen en cuenta a la hora de ejecutar el programa. Son muy útiles para luego repasar las prácticas de cara al examen. Te recomendamos que vayas apuntando con # todo lo que vaya comentando Arturo.

```
# Suma de dos valores y almacenarlo en una variable
ii = 3+5
# Mostrar el resultado escribiendo la variable
ii
```

3º Es muy importante tener en cuenta que los **decimales** hay que ponerlos con puntos, y no con comas. Las comas se usan para separar valores en vectores.

Si ponemos comas nos aparece lo siguiente en la consola:

```
a = 3
b = 4,5
S = a + b
S
Error: inesperado ',' in "b = 4,"
Error: objeto 'b' no encontrado
Error: objeto 'S' no encontrado
```

Sin embargo, si ponemos puntos efectúa la suma sin problema:

```
> a = 3
> b = 4.5
> S = a + b
> S
[1] 7.5
```

4º Si escribimos **help("algo")** en la consola, nos redirige directamente a una página donde nos explican la función o lo que hayamos metido dentro del paréntesis. ¡¡Es muy importante poner las comillas!!

help("for")



<http://127.0.0.1:30290/library/base/html/Control.html>

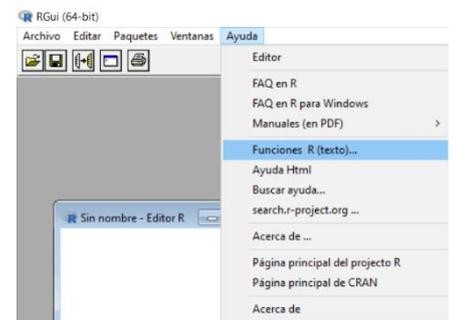
Description  
These are the basic control-flow constructs of the R language. They function in much the same way as control statements in any Algol-like language. They are all reserved words.

Usage  
if (cond) expr  
else cond2 expr2 else cond3 expr3  
Do {expr} until cond  
repeat {expr}  
while (cond) {expr}  
break  
next

Arguments  
cond  
cond2  
cond3  
expr  
expr2  
expr3  
expr  
cond  
expr

Details  
break breaks out of a while, until, repeat, or for loop; control is transferred to the first statement outside the inner-most loop. next halts the processing of the current iteration and advances the looping index. Both break and next apply only to the innermost of nested loops.  
Note that it is a common mistake to forget to put braces { ... } around your statements, e.g., while (1) { } or do { } { }. In particular, you should not have a newline between ; and a line to avoid a syntax error in entering a if ...

También podemos acceder a ayuda de forma más sencilla y sin recordar comillas ni paréntesis:



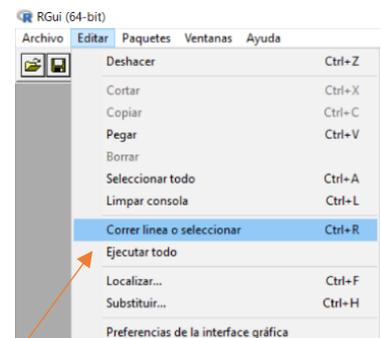
5º Te preguntarán qué es eso de un **script y la consola**. Pues es muy sencillo. La principal diferencia es que en el script podemos modificar todo lo que hemos escrito las veces que queramos. Sin embargo, en la consola no es posible, y hay que escribir de nuevo todo el programa. Por eso, mejor escribir nuestro programa en un script nuevo.

6º Para que el programa **ejecute** una línea o el script entero tenemos tres formas:



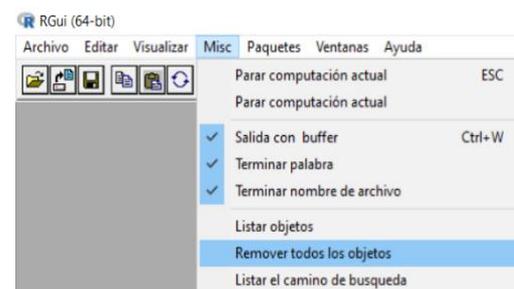
➤ Seleccionar la parte que necesitamos y darle al símbolo:

➤ Seleccionar la parte que necesitamos y darle a editar → correr línea o seleccionar.

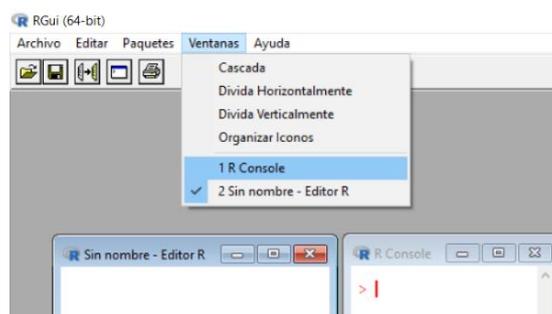


➤ También podemos ejecutar todo el script dándole a editar → ejecutar todo

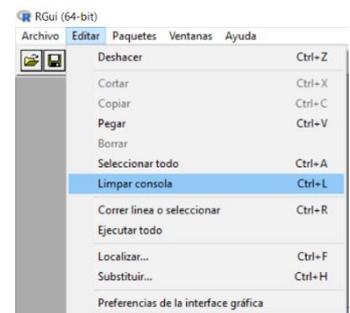
7º Al principio de cada práctica es útil **borrar** los **datos antiguos almacenados** en la consola. Puede que hayamos asociado el nombre “aa” a un objeto, y ahora queremos meter otro objeto distinto en esa dirección (pero ya está ocupada). Para remover los objetos → Misc → remover todos los objetos.



8º Cuando tenemos varias ventanas abiertas (consola, scripts, gráficos...) a veces desaparece una ventana (se minimiza) Para ver qué **ventanas** tenemos **abiertas** → ventanas y tienen que aparecer todas las que hay en ese momento.

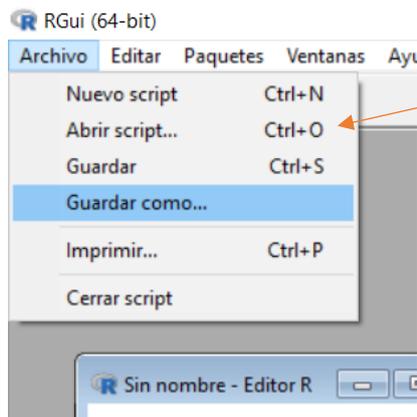


9º Para **vaciar la consola** si hemos llenado de datos la misma y nos resulta más cómodo comenzar a ver el proceso desde 0 le damos a editar → limpiar consola.



10º Por último, una de las ideas más importantes es cómo **guardar y abrir los scripts**.

➤ Para guardarlos:



➤ Para abrirlos

Esperamos que te haya sido útil el documento y puedas empezar este camino de una forma más sencilla. Recuerda que tienes que ir poco a poco. Si lo dejas todo para el final se acumulará todo el trabajo y será muy agobiante.