Introducción a R

Modelos no paramétricos y de regresión

Enrique Reyes

29 de enero de 2019

Introducción

R es un programa muy potente para el análisis estadístico de datos, pero no sólo se destaca en esta rama del conocimiento, tiene muchas aplicaciones por ejemplo en: Geografía, Análisis Numérico, Data Science, Big Data, entre otras. Esto se debe principalmente a que es un lenguaje sencillo de contribución libre, es decir pertenece al sistema *GNU* por lo tanto constantemente hay actualizaciones, nuevas funciones y corrección de errores, sigue la idea de que el conocimiento es de todos, el conocimiento no es estático, por lo tanto se deben compartir los nuevos hallazgos para seguir mejorando.

R se ha popularizado, cada vez son más las empresas, organizaciones e institutos que se acercan a este programa, por ser un software libre de multiplataforma (compatible con Windows, Mac, Linux), con muchas paqueterías para el análisis de datos, además de tener una interfaz gráfica amigable *R Studio*.

Instalación

Como se mencionó, *R* es libre y multiplataforma, por ende no importa el sistema operativo que el usuario tenga, este se puede descargar e instalar de manera gratuita; la versión a utilizar es 3.5.2, la dirección para descargar el programa es:

- *Windows*
- *Mac*
- *Linux* (*Debian, Redhat, Suse, Ubuntu*)

Pero como el interfaz gráfico no es muy amigable, utilizaremos *R* por medio de *R Studio*, ya que este es más visual, más dinámico, se hace más ágil la manipulación de instrucciones, la versión 1.1.463 es la más actual, esta interfaz también es multiplataforma y gratuita, se puede descargar desde aquí.

Cabe mencionar dos cosas: es necesario instalar en este orden los programas, si se instala primero *R Studio* el programa marcará muchos errores por la ausencia del lenguaje principal; y las versiones base no son necesariamente serán fijas, haby que estar al tanto de las nuevas actualizaciones e istalarlas de manera oportuna, muchas paqueterías dejan de ser estables con versiones anteriores del sistema.

Primeros pasos en *R*

R Studio abre por default 4 ventanas:

1. Ventana de scripts: en esta sección se crearán y se gardarán los códigos.
2. Ventana de consola: aquí se mostrarán los resultados de los códigos, también se pueden escribir comandos pero no se guardarán.
3. Ventana de variables: en esta sección se mostrarán todos los objetos que se declaren en el código, es de gran utilidad conocer el nombre de las variables, esto evitará la eliminación o sobreescritura de alguna variable, así mismo sabremos que objetos eliminar cuando estos dejen de servir para nuestro análisis. Adicionalmente en esta sección se almacenará el historial del código.
4. Ventana gráfica: en está sección, se mostrarán las gráficas que se vayan haciendo, no sólo eso, también estarán las paqueterías, la ayuda y los documentos que están dentro del fichero predefinido (Por default en Windows esta carpeta es Documentos).

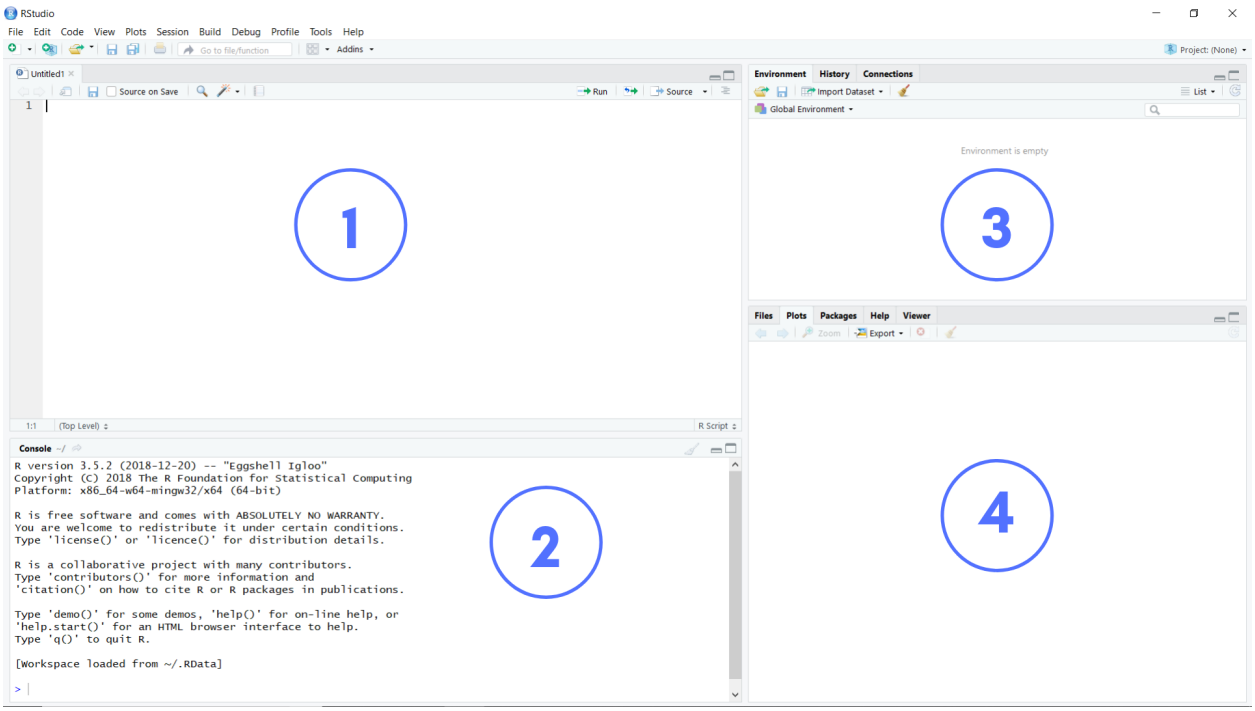


Figure 1: Interfaz Gráfica de R Studio

Directorio de trabajo

Una de las primeras cosas que debemos saber es dónde estamos trabajando, en qué carpeta se guardará todo nuestro contenido. Entonces, para saber cuál es la carpeta de trabajo, debemos poner en la consola el siguiente comando:

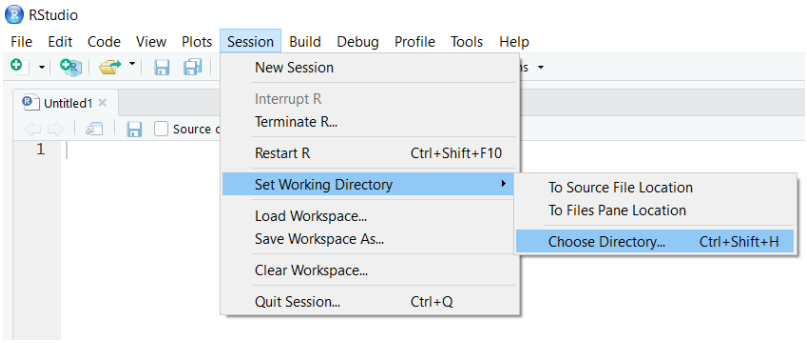
```
getwd()
```

```
## [1] "C:/Users/Enrique Seyer/Dropbox/MNPR"
```

Para cambiar el directorio de trabajo necesitamos obtener la nueva carpeta y la funcion *setwd()*, entonces si queremos cambiar la dirección a una carpeta llamada Clases, la instrucción será la siguiente:

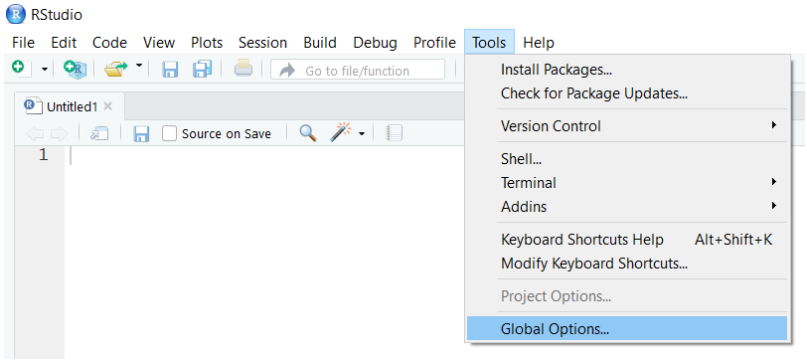
```
setwd("D:/Clases")
```

Para hacer este mismo cambio pero de manera visual, tenemos que ir a la pestaña *Session*, ir a *Set working Directory*, y seleccionar *Choose Directory*:

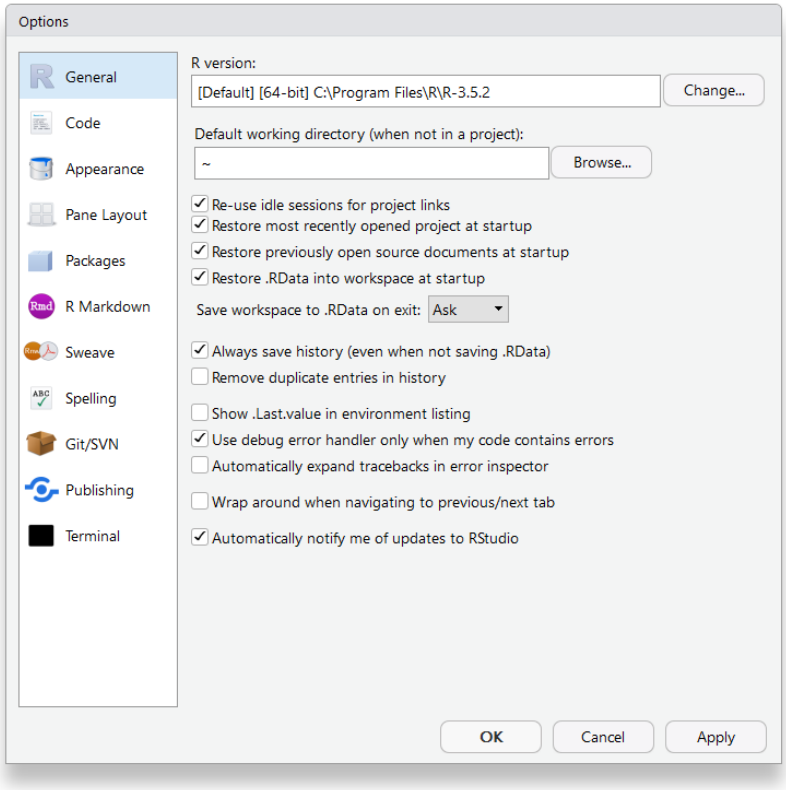


El inconveniente de seleccionar el directorio de esta forma es que en cuanto se cierre el programa se borrará esta configuración y se tendrá que hacer este procedimiento una vez más cuando se vuelva abrir el programa,

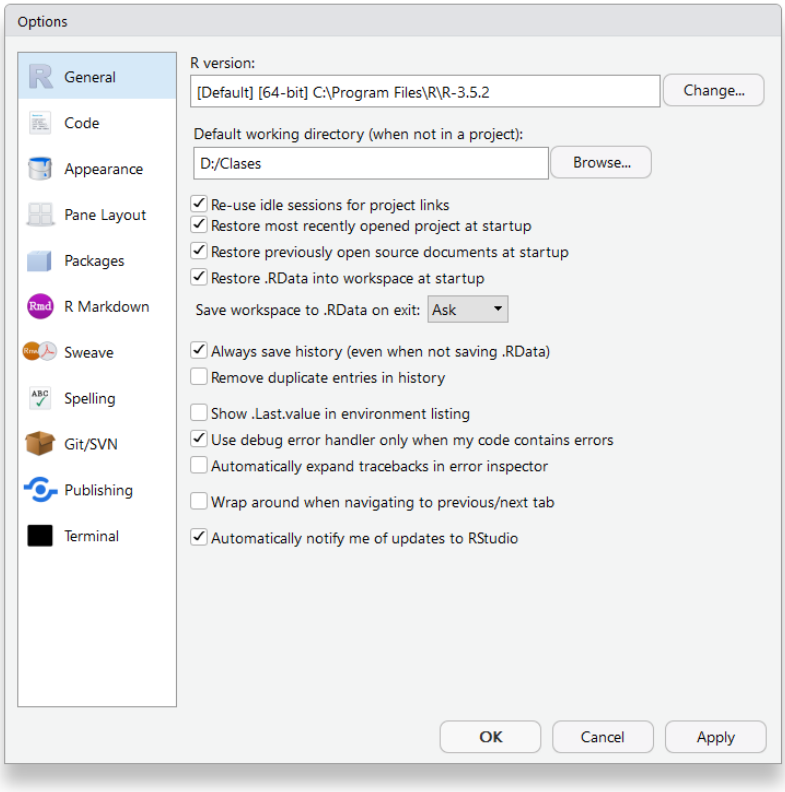
si se trabajará constantemente en ese directorio se recomienda definir el directorio por *default*, esto se hace seleccionado la pestaña *Tools*, ir a la última opción *Global Options*



después se abrirá una nueva ventana ahí se debe seleccionar la opcion *General* y en la opción *Default working directory*, pulsar el botón *Browse...*

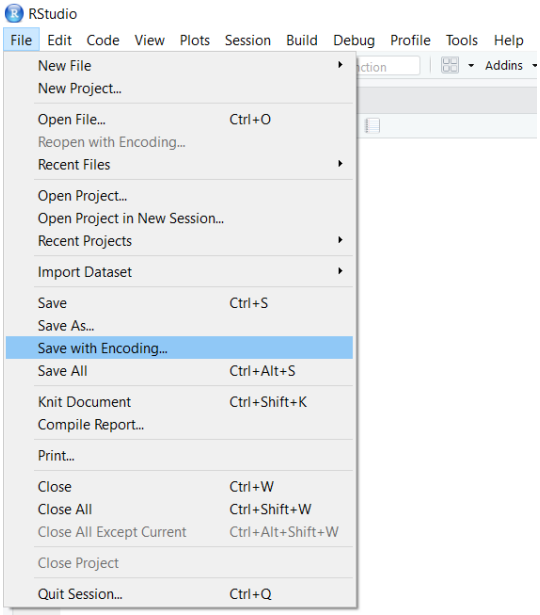


aquí se buscará el directorio deseado, una vez encontrado, se debe presionar el botón *Ok* y se guardarán los cambios, de esta forma siempre que se abra R, esta será la carpeta de trabajo.



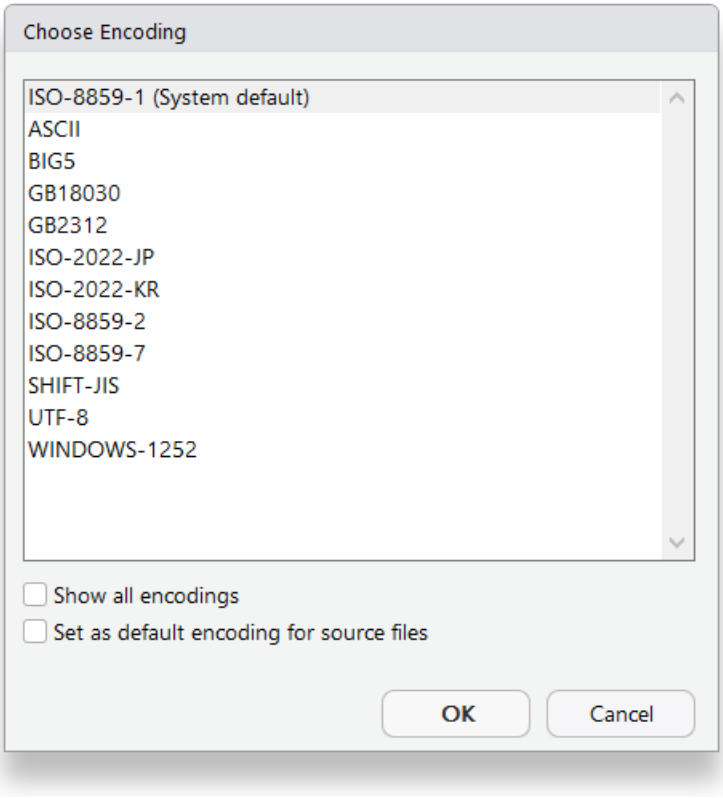
Codificación

Hay varios tipos de codificación, dentro de los más conocidos están **ASCII**, **UTF-8**, **ISO 8859-1**; la codificación universal es **ASCII**, de los más utilizados en la actualidad es **UFT-8** y el recomendado para la codificación latina es **ISO 8859-1**, para definir alguna codificación tenemos que ir a la ficha *File* y seleccionar la opción *Save with Encoding...*



una vez seleccionada esta opción aparecerá una ventana con algunas recomendaciones de codificación o se

puede ver todas las opciones disponibles, aquí seleccionará la más conveniente, para guardar estos cambios, sólo necesitamos oprimir el botón de *Ok*.



Metacomandos

Windows/Linux	*Mac*	Acción
Ctrl+Shift+H	Cmd+Shift+H	Cambiar directorio de trabajo
Ctrl+Shift+N	Cmd+Shift+N	Nuevo script
Ctrl+Shift+R	Cmd+Shift+R	Inserta una nueva sección (script)
Ctrl+O	Cmd+O	Abrir un documento
Ctrl+S	Cmd+S	Guardar el script actual
Ctrl+Alt+S	Cmd+Alt+S	Guarda todos los scripts abiertos
Ctrl+L	Cmd+L	Limpia la consola
Ctrl+Enter	Cmd+Enter	Corre una línea del script
Ctrl+Shift+P	Cmd+Shift+P	vuelve a correr lineas anteriores
Ctrl+W	Cmd+W	Cierra el script actual
Ctrl+Shift+W	Cmd+Shift+W	Cierra todos los scripts
ctrl+3	Cmd+3	Abre la ventana de ayuda
Ctrl+4	Cmd+4	Muestra el historial
Ctrl+5	Cmd+5	Abre la ventana del folder de trabajo
Ctrl+6	Cmd+6	Muestra la ventana de gráficas
Ctrl+7	Cmd+7	Muestra los paquetes
Ctrl+8	Cmd+8	Muestra las variables almacenadas
Ctrl+Shift+K	Cmd+Shift+K	Crea un archivo con la paquetería *Knit*

Primeros comandos

```
#ayuda/documentación
help()
help.start()
?
#salida
q()
#lista de objetos
ls()
#eliminar objetos especificos
rm()
#guardar variables y el historial
save()
#carga algun historial especifico
load()
```

Comandos de navegación en el disco duro

```
#dirección actual de trabajo
getwd()
#crea carpetas
dir.create()
#muestra las carpetas en el directorio
list.files()
#nueva dirección de trabajo
setwd()
#crea un nuevo archivo
file.create()
#elimina archivos
file.remove()
#copia archivos
file.copy()
```

Operaciones básicas

```
#sumas
5+5
```

```
## [1] 10
```

```
#restas
8-3
```

```
## [1] 5
```

```
#multiplicaciones
6*5
```

```
## [1] 30
```

```
#divisiones
7/5
```

```
## [1] 1.4
```

```r
#modulo
8%%2
```

```
## [1] 0
```

```r
#parte entera de la division
6%/%3
```

```
## [1] 2
```

```r
#exponentes
4^5
```

```
## [1] 1024
```

```r
2**3
```

```
## [1] 8
```

```r
#raíz cudrada
sqrt(144)
```

```
## [1] 12
```

```r
#signo
sign(1)
```

```
## [1] 1
```

```r
sign(0)
```

```
## [1] 0
```

```r
sign(-1)
```

```
## [1] -1
```

```r
sign(-10)
```

```
## [1] -1
```

```r
sign(10)
```

```
## [1] 1
```

```r
#valor absoluto
abs(1)
```

```
## [1] 1
```

```r
abs(-1)
```

```
## [1] 1
```

```r
#función piso
floor(1.4)
```

```
## [1] 1
```

```r
floor(1.9)
```

```
## [1] 1
```

```r
#función techo
ceiling(1.4)
```

```
## [1] 2
```

```
ceiling(1.9)
```

```
## [1] 2
```

```
#truncar
trunc(1.8564)
```

```
## [1] 1
```

```
#redondear
round(1.8564,3)
```

```
## [1] 1.856
```

```
#significancia
signif(1.574309463,4)
```

```
## [1] 1.574
```

```
#logaritmo natural
log(exp(1))
```

```
## [1] 1
```

```
#logaritmo base 10
log10(10)
```

```
## [1] 1
```

```
#logaritmo base 2
log2(2)
```

```
## [1] 1
```

```
#logaritmo base 1.71828
log1p(1.718282)
```

```
## [1] 1
```

```
#logaritmo genérico, cualquier base
log(3,3)
```

```
## [1] 1
```

```
#exponencial
exp(2)
```

```
## [1] 7.389056
```

```
#factorial
factorial(5)
```

```
## [1] 120
```

```
#pi
pi
```

```
## [1] 3.141593
```

```
#Seno
sin(2*pi)
```

```
## [1] -2.449213e-16
```

```r
sinpi(2)
```

```
## [1] 0
```

```r
#coseno
cos(2*pi)
```

```
## [1] 1
```

```r
cospi(2)
```

```
## [1] 1
```

```r
#tangente
tan(2*pi)
```

```
## [1] -2.449294e-16
```

```r
tanpi(2)
```

```
## [1] 0
```

```r
#arcocoseno
acos(1)
```

```
## [1] 0
```

```r
#arcoseno
asin(0)
```

```
## [1] 0
```

```r
#arcotangente
atan(1)
```

```
## [1] 0.7853982
```

```r
#funciones hiperbólica
sinh(1)
```

```
## [1] 1.175201
```

```r
cosh(1)
```

```
## [1] 1.543081
```

```r
tanh(1)
```

```
## [1] 0.7615942
```

```r
acosh(2)
```

```
## [1] 1.316958
```

```r
asinh(2)
```

```
## [1] 1.443635
```

```r
atanh(2)
```

```
## Warning in atanh(2): Se han producido NaNs
```

```
## [1] NaN
```

```r
atanh(1)
```

```
## [1] Inf
```

```
atanh(1/2)
```

```
## [1] 0.5493061
```

```
#complejos
5+3i
```

```
## [1] 5+3i
```

```
1i*1i
```

```
## [1] -1+0i
```

```
1i
```

```
## [1] 0+1i
```

```
Re(1+3i)
```

```
## [1] 1
```

```
Re(2i)
```

```
## [1] 0
```

```
Im(3+1i)
```

```
## [1] 1
```

```
Im(2)
```

```
## [1] 0
```

```
Mod(1i)
```

```
## [1] 1
```

```
Mod(1+1i)
```

```
## [1] 1.414214
```

```
Mod(7)
```

```
## [1] 7
```

```
Arg(3i)
```

```
## [1] 1.570796
```

```
Arg(3)
```

```
## [1] 0
```

```
Arg(2+7i)
```

```
## [1] 1.292497
```

```
Conj(1i)
```

```
## [1] 0-1i
```

```
Conj(4)
```

```
## [1] 4
```

```r
Conj(4+8i)
```

```
## [1] 4-8i
```

Tipos de variables

Numéricas

```r
#objeto simple numérico
a<-1;a
```

```
## [1] 1
```

```r
b=4;b
```

```
## [1] 4
```

```r
### Todas las operaciones anteriores son objetos numéricos
```

Carácter

```r
#siempre debe ir entre comillas los archivos de tipo texto
#c1<-Hola #error
z<- "Hola mundo"; z
```

```
## [1] "Hola mundo"
```

```r
a1="Hola"; a1
```

```
## [1] "Hola"
```

```r
b1="a todos"; b1
```

```
## [1] "a todos"
```

```r
#Pega el contenido de dos cadenas de caracter
paste(a1,b1)
```

```
## [1] "Hola a todos"
```

```r
c1="¿Cómo están?"; c1
```

```
## [1] "¿Cómo están?"
```

```r
paste(a1,b1,c1)
```

```
## [1] "Hola a todos ¿Cómo están?"
```

```r
#Cuenta el número de caracteres
nchar(b1)
```

```
## [1] 7
```

```r
#Busca una frase dentro de una cadana de caracteres
grep("todos",b1)
```

```
## [1] 1
```

```r
#Sólo mayusculas
toupper(b1)
```

```
## [1] "A TODOS"
```

```r
#sólo minusculas
tolower(a1)
```

```
## [1] "hola"
```

```r
#cambia el contenido de una cadena
sub("todos","nadie",b1)
```

```
## [1] "a nadie"
```

Lógicos

```r
#objeto lógico
e<-TRUE; e
```

```
## [1] TRUE
```

```r
f=F; f
```

```
## [1] FALSE
```

```r
#validaciónes
#menor que
6<9
```

```
## [1] TRUE
```

```r
7<2
```

```
## [1] FALSE
```

```r
#mayor
7>5
```

```
## [1] TRUE
```

```r
#menor o igual que
5<=7
```

```
## [1] TRUE
```

```r
5<=5
```

```
## [1] TRUE
```

```r
5<=1
```

```
## [1] FALSE
```

```r
#mayor que
10>1
```

```
## [1] TRUE
```

```r
#igual
10==11
```

```
## [1] FALSE
```

```r
#diferente
2!=3
```

```
## [1] TRUE
```

```
#se verán más adelante
#y
#ÿ
#o
#|
#Verificación de tipo de objeto
is.numeric("42")
```

```
## [1] FALSE
```

```
is.numeric(13)
```

```
## [1] TRUE
```

```
is.integer(1.78)
```

```
## [1] FALSE
```

```
is.logical(30<0)
```

```
## [1] TRUE
```

```
is.logical(F)
```

```
## [1] TRUE
```

```
is.character(4)
```

```
## [1] FALSE
```

```
is.character("¿Qué tal?")
```

```
## [1] TRUE
```

Una observación interesante es que R tambien considera de tipo numerico a las variables lógicas, en donde $FALSE \equiv 0$ y $TRUE \equiv 1$, se pueden operar estas variables sin complicaciones

```
TRUE+15
```

```
## [1] 16
```

```
exp(FALSE)
```

```
## [1] 1
```

```
pi**(-TRUE)
```

```
## [1] 0.3183099
```