

PRACTICA.1

Nelson de Jesus Magaña Godinez

28/7/2022

INTRODUCCION AL ENTORNO

- Revisar las demostraciones o demos siguientes: `demo(graphics)`, `demo(image)`, `demo(persp)` y `demo(plotmath)`, `demo(intervals)`, `demo(lattice)` (en los últimos dos es necesario cargar previamente el paquete `lattice`)

3. Con `objects()` o `ls()` puede listar los objetos creados en el espacio de trabajo o memoria (Workspace), al ejecutar los demos anteriores.

Luego puede eliminar o remover los objetos con `rm(list=ls())` o `remove(list=objects())`

Note: que también se pueden listar y remover objetos desde los menús. Seleccione para esto el Menú Misc, y dentro de el seleccione listar o remover objetos.

4. Crear en la raíz del disco duro “C:/” o en “Mis Documentos” una carpeta o directorio con su nombre para guardar sus prácticas.

5. Ver un listado de las carpetas y archivos contenidos en un directorio utilizando las funciones: • `dir()`, por ejemplo, `dir("C:/", pattern = "1", full.names=TRUE)`

```
dir("C:/", pattern = "[a-p]", full.names = TRUE)
```

```
## [1] "C:/bootTel.dat" "C:/found.000" "C:/hiberfil.sys" "C:/hp"
## [5] "C:/pagefile.sys"
```

Note que la instrucción “2” le indica a R que liste los archivos que empiezan con letras de la “a” hasta la “p”.

- O también `list.files()`, por ejemplo, `list.files("C:/", pattern = NULL, all.files = TRUE, full.names = FALSE)`

```
list.files("C:/", pattern = NULL, all.files = TRUE, full.names = FALSE)
```

```
## [1] "$AV_ASW" "$GetCurrent"
## [3] "$RECYCLE.BIN" "$SysReset"
## [5] "$WINDOWS.BT" "$WinREAgent"
## [7] "bootTel.dat" "Documents and Settings"
## [9] "found.000" "hiberfil.sys"
```

¹a-p
²a-p

```
## [11] "hp" "Intel"
## [13] "pagefile.sys" "PerfLogs"
## [15] "Program Files" "Program Files (x86)"
## [17] "ProgramData" "Recovery"
## [19] "swapfile.sys" "SWSetup"
## [21] "System Volume Information" "SYSTEM.SAV"
## [23] "SysWOW64" "Users"
## [25] "Windows" "Windows10Upgrade"
## [27] "xampp"
```

Note que con la instrucción anterior se muestran todos los archivos visibles y no visibles (ocultos y protegidos por el sistema).

6. R utiliza el directorio de trabajo para leer y escribir archivos. Para saber cual es este directorio puede utilizar la función `getwd()` (*get working directory*). Para cambiar el directorio de trabajo, se utiliza la función `setwd()`; por ejemplo, `setwd("C:/Curso R2012")`.

```
getwd()
```

```
## [1] "C:/Users/pc 1/Desktop/PAQUETE R/PRACTICAS-_1"
```

Es necesario proporcionar la ruta ('path') completa del archivo si este no se encuentra en el directorio de trabajo de R, el cual por defecto es "C:/Archivos de programa/R/R-2.13.1".

7. Ejemplos de cálculo numérico en la Consola de R (R-Console)

Ejemplo 1. Encontrar el resultado de operar: 2 más 10 por 3 entre 5 Escriba en la Consola de R: $2+10*3/5$ y oprima la tecla ENTER

```
2+10*3/5
```

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

Note que en R se respecta el mismo orden de preferencia de la mayoría de los lenguajes de programación, la multiplicación y la división tienen prioridad a la suma y resta.

Ejemplo 2. Encontrar el resultado de operar: 3 elevado a la potencia 100 3^{100} o también `format(3^100, sci = FALSE)` `Sci=FALSE` le indica a R que muestre todos los dígitos del resultado, de lo contrario (`Sci=TRUE`) solamente se mostrará la representación científica.

```
3^100 #El resultado lo muestra en notacion cientifica.
```

```
## [1] 5.153775e+47
```

```
format(3^100, scientific = FALSE)
```

```
## [1] "515377520732011324202202224420402268886864624842"
```

```
format(3^100, scientific = TRUE)
```

```
## [1] "5.153775e+47"
```

Ejemplo 3. Encontrar el resultado anterior con 15 cifras decimales y guardarlo en la variable `y` `y <- format(3^100, digits = 15);y` o `y = format(3^100, digits = 15)`

```
y<-format(3^100, digits = 15);  
y
```

```
## [1] "5.15377520732011e+47"
```

```
y=format(3^100, digits = 15);  
y
```

```
## [1] "5.15377520732011e+47"
```

Note que en R, la asignación de valores a una variable puede hacerse con “=” o con “<-”.

Ejemplo 4. Redondear el valor de π a 4 dígitos decimales `round(pi, 4)` Aplique las funciones: `trunc(pi)`, `floor(pi)` y `ceiling(pi)`

```
round(pi, 4)
```

```
## [1] 3.1416
```

```
trunc(pi)
```

```
## [1] 3
```

```
floor(pi)
```

```
## [1] 3
```

```
ceiling(pi)
```

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

Ejemplo 5. Guardar en la variable `n` el valor 150 y luego calcular el valor de `n` `n = 150 factorial(n)`

```
n<-50  
n
```

```
## [1] 50
```

```
factorial(n)
```

```
## [1] 3.041409e+64
```

Ejemplo 6. Operar el complejo $(2+3i)$ elevado a la potencia 10 $(2+3i)^{10}$ o también `format((2+3i)^10, sci = TRUE)`

```
(2+3i)^10
```

```
## [1] -341525-145668i
```

```
format((2+3i)^10, scientific = TRUE)
```

```
## [1] "-3.41525e+05-1.45668e+05i"
```

Ejemplo 7. Calcular la integral entre 0 y π de la función $\text{Sen}(x)$

```
f=function(x) {sin(x)}  
integrate(f, lower = 0, upper = pi)
```

```
## 2 with absolute error < 2.2e-14
```

TRABAJANDO CON SCRIPT

A medida que estemos realizando un trabajo de pequeña, mediana o de gran complejidad, será muy útil manejar todas las entradas que solicitemos a R en un entorno donde podamos corregirlas, retocarlas, repetirlas, guardarlas para continuar el trabajo en otro momento, con otros datos, etc. Esta es la función del editor de R, a los archivos creados en este editor se les conoce como Script. Es posible incluir comentarios que R no leerá si utilizamos líneas que comiencen con el carácter `#` (o en cualquier parte de la línea). Por el contrario, si escribimos cualquier orden no antecedida de `#` R (sin importar en que parte se encuentre) lo reconocerá como instrucciones que deben ejecutarse.

1. Crear un script o guión.
2. Realizar en el script los siguientes cálculos numéricos.

```
2*(3+4)^2
```

```
## [1] 98
```

```
sqrt(16) #rais cuadrada de 16
```

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

```
abs(-97.6) #abs(x) calcula el valor absoluto de x
```

```
## [1] 97.6
```

```
x = 4 # almacena el valor de 4 en la variable x
x # Muestra el contenido de la variable x
```

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

```
sqrt(x)-3/2
```

```
## [1] 0.5
```

```
p<-(4>8)
p
```

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

```
q <- -6+4<3 && 4!=10
q
```

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

```
r = -6+4>3 || 4==10
r
```

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

```
t<- !r
t
```

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

```
sin(pi/2)
```

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

```
(y=cos(pi)) # Los primeros parentesis permiten ver el calculo de la variable y sin necesidad de llamarla
```

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

Calcula el Logaritmo natural de 3

```
log(3)
```

```
## [1] 1.098612
```

Calcula el logaritmo base 10 de 8

```
log10(8)
```

```
## [1] 0.90309
```

La sintaxis general es: $\log_b(x, base)$

```
logb(16, 7)
```

```
## [1] 1.424829
```

$\exp(x)$ Calcula la función exponencial e^x

```
exp(1)
```

```
## [1] 2.718282
```

Después de digitar el script, marque con el ratón las líneas 5, 6 y 7, ejecútelas oprimiendo el botón derecho del ratón, y luego eligiendo la opción “Correr línea o seleccionar”. También puede ejecutar una línea posicionando el cursor sobre cualquier lugar de ella y oprimiendo simultáneamente las teclas Ctrl y R.

3. Ejecute todas las líneas o instrucciones del script.
4. Guarde el script en su directorio de trabajo, puede llamarle Script-Practica01, el programa le da automáticamente la extensión .R

Nota: para guardar el script hay que tener la ventana activa y el Menú: Archivo—>Guardar

5. Salga del programa R, ejecutando la función `q()` o desde Menú: Archivo—>Salir

```
#q()
```

6. Entre nuevamente al R, recupere el archivo donde guardó el script, como se muestra en la figura, y ejecute algunas instrucciones.

NOTA: Si se escribe el nombre de la función sin los paréntesis, R mostrará el código de algunas funciones. Por ejemplo, `ls`