

Sesión 1: Una introducción a R

Métodos estadístico de investigación: Introducción a R y Rstudio

000R Team

2016/17

- 1 Introducción
- 2 R: el lenguaje
- 3 Trabajando con R
- 4 Preguntas

Introducción

Objetivos de la sesión

Conocer y comprender

1 Conocer

- la principales ventajas de R
- el funcionamiento básico de la terminal de R
- los principales elementos de la sintaxis de R
- el procedimiento básico de trabajo con R

2 Comprender

- el fundamento de la sintaxis de R
- el procedimiento de trabajo
- los mensajes de error del sistema

¿Qué es R?

Definición

- Permite el almacenamiento, manejo y tratamiento estadístico de los datos
- [R] se desarrolló sobre una idea de R Becker, J Chambers y A Wilks
- *lingua franca* de la estadística y los aspectos cuantitativos de numerosos campos del conocimiento:
 - biología (ecología, genética, filogenia...), farmacología, ...
 - economía, finanzas, ...
 - Química, física,
 - optimización, etc.

Qué es R

Vídeo traducido

<http://gauss.inf.um.es/videos/whatsR.webm>

Unas fotos de familia

- Interfaces

`http://www.statmethods.net/interface/guis.html`

- Instalación
- rstudio

rstudio

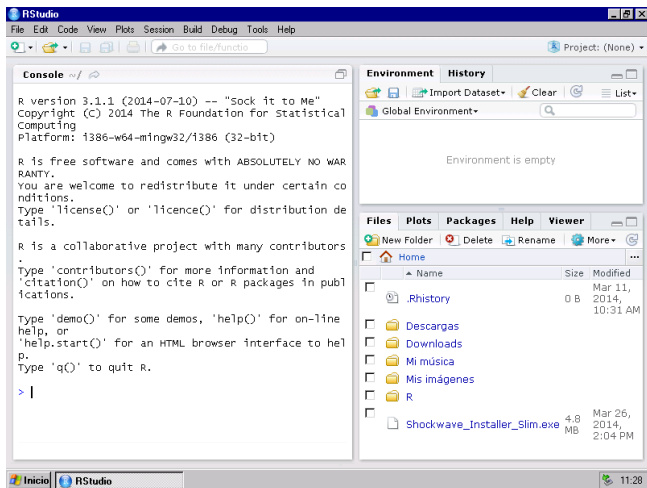


Figure 1.

rstudio: la terminal

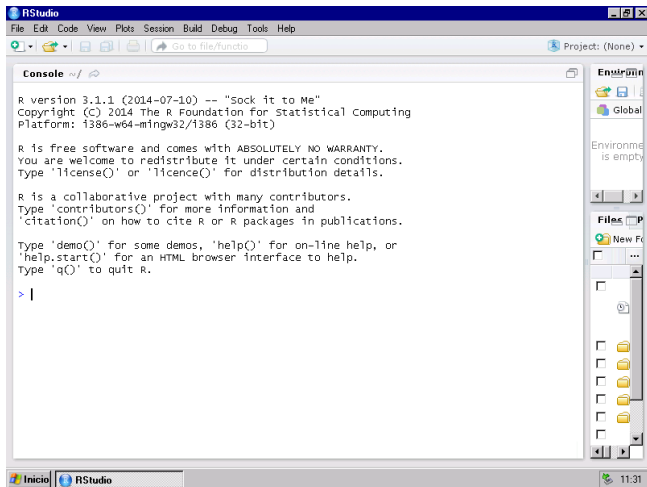


Figure 2:

¿Qué tiene R que tanto nos gusta?:

- Es libre. licencia GNU, → utilizar y ¡mejorar!
- Es multiplataforma: Linux, Windows, Mac, iPhone...
- Se puede analizar en R cualquier tipo de datos.
- Es potente. Es muy potente.
- Capacidad gráfica. Difícilmente es superada por ningún otro paquete estadístico.
- Compatible con 'todos': csv, xls, sav, sas...
- Es ampliable, si quieres añadir algo: ¡empaquetalo!
- Hay miles de técnicas estadísticas implementadas, cada día hay más.

Importancia de la comunidad

- R aumenta su capacidad con la colaboración de los usuarios
 - 1998 unas 200 librerías
 - 2011, octubre, más de 3300
 - ¿Hoy cuantas?

Sobre la notación y la tipografía

Comunicación con un autómata

- El autómata carece de inteligencia
- R hace lo que se le pide, no lo que se quiere
- En una conversación deben respetarse las reglas de comunicación
- Las reglas tipográficas ayudan a simplificar

De la escritura

- El manejo del teclado es muy importante
- Atajos de teclado, *hotkeys* y *shortcuts*
- Sensibilidad a mayúsculas (*case sensitive*): no es lo mismo 'A' que 'a'
- El uso del tabulador para autocompletado

De la pantalla

- Intercomunicación: mensajes de respuesta
- Errores: *Warning*
- Errores: *Fatal error*
- Malditos errores: *Syntax error*

La terminal de R

- Bienvenida
- Expresiones para R
- El *prompt*
 - >
 - +
- La despedida

rstudio

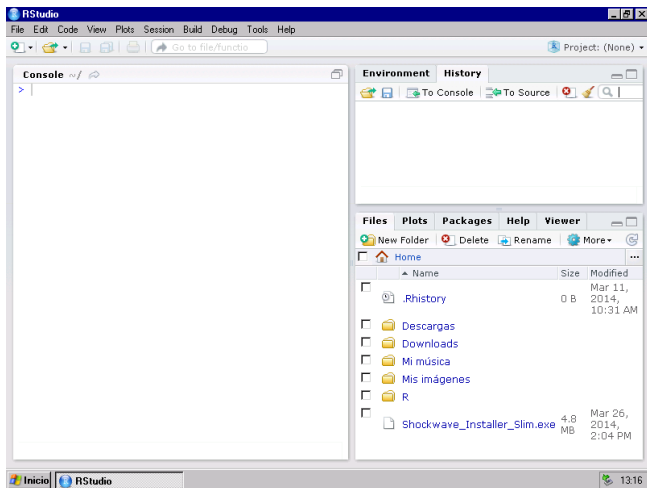


Figure 3:

rstudio: Usando el tabulador

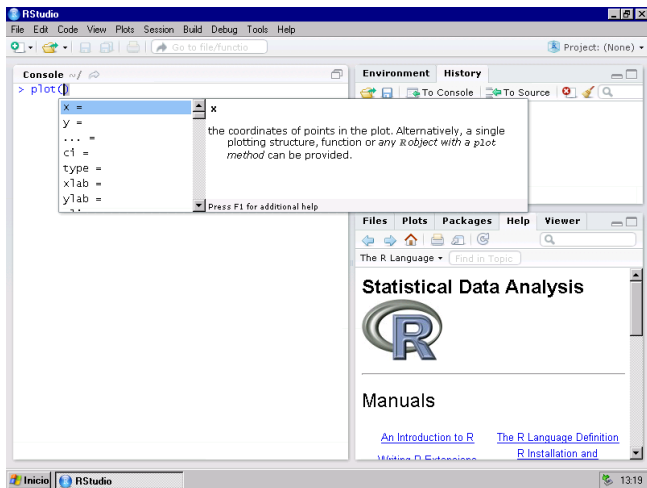


Figure 4:

Elección del directorio de trabajo

rstudio: Entrada Session

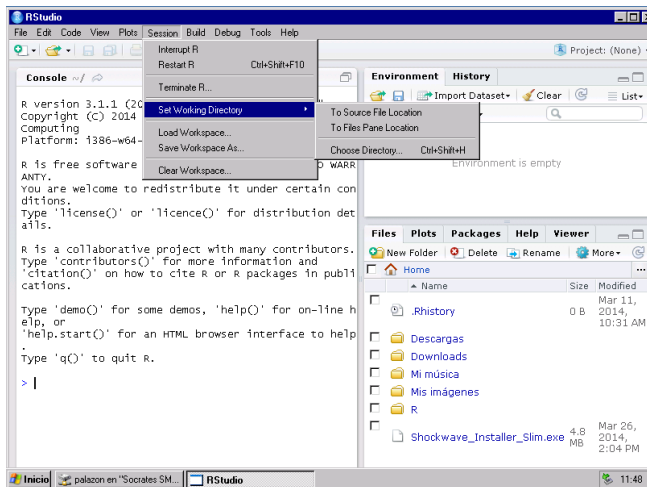


Figure 5:

rstudio: Localizando el directorio de trabajo

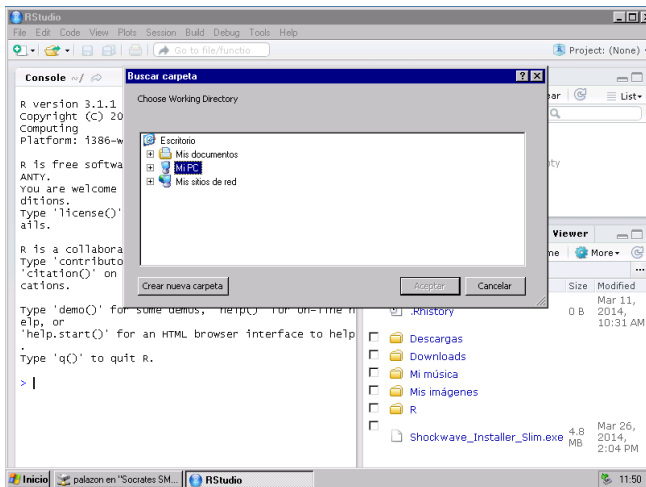


Figure 6:

rstudio: Seleccionando la unidad sócrates

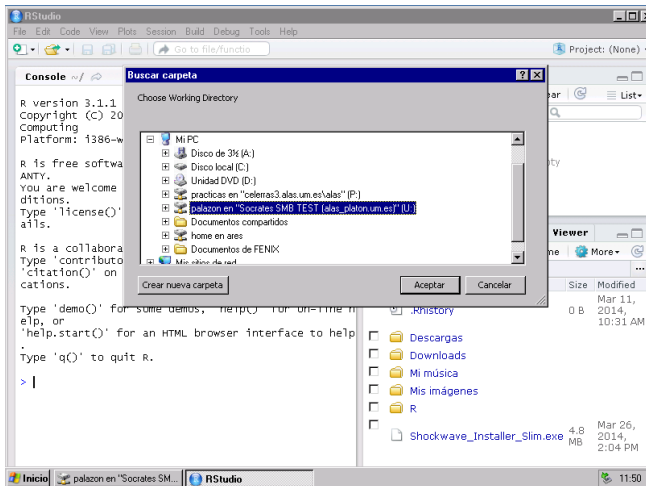


Figure 7:

rstudio: directorio para el seminario

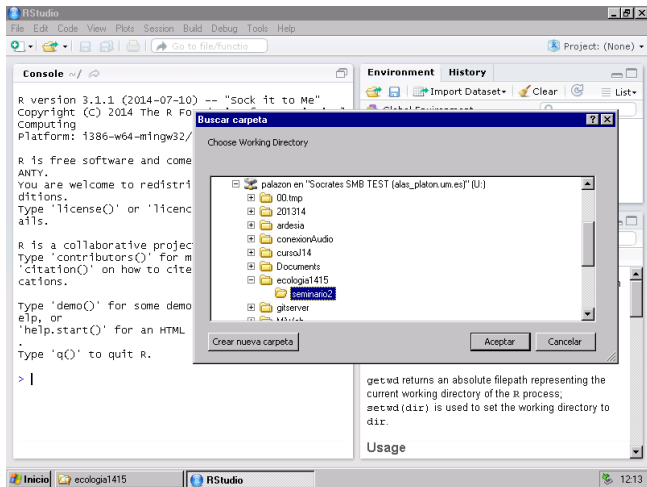


Figure 8:

rstudio: ¡Una expresión!

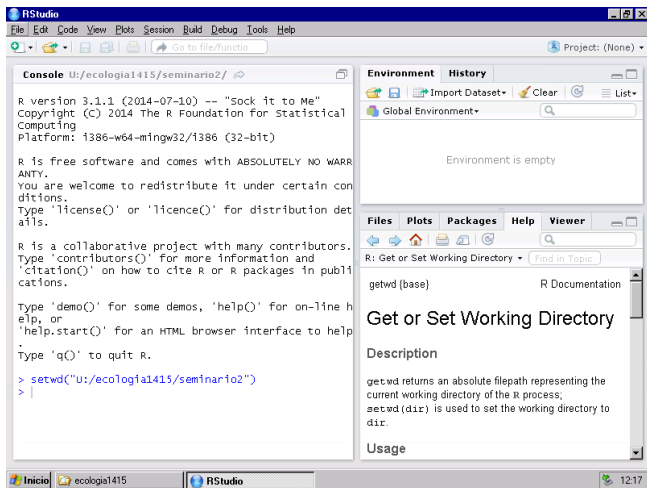


Figure 9:

R: el lenguaje

Sobre la notación

Reglas básicas de sintaxis R

- Reglas sintácticas
 - ① R evalúa expresiones
 - ② El lenguaje es sensible a mayúsculas
 - ③ Pueden utilizarse espacios entre elementos de sintaxis a discreción: `sin(x+b)` es igual que `sin (x + b)`
 - ④ Cada expresión se escribe en al menos una línea
 - ⑤ Dos o más expresiones puede utilizar una línea separándolas por el signo `' ; '`
- En R, donde entra un valor puede entrar una expresión
- Regla de reuso
- ESC una tecla para huir, abortar, cortar,...

Notación matemática y sintaxis de R

Matemáticas	Expresión en R
$x = 3$	<code>x <- 3</code>
$\sin \alpha$	<code>sin(alpha)</code>
$\log_{10}(x)$	<code>log(x, 10)</code>
v_i	<code>v[i]</code>
$\sum_{i=1}^n v_i$	<code>sum(v)</code>

Elementos de R

Valores

- Enteros: 3
- Reales: $1.8e+12$ ($1.8 \cdot 10^{12}$)
- Complejos: $0+1i$ ($\sqrt{-1}$)
- Carácter: "rojo"
- Perdidos: NA
- No números: NaN ($\log(0)$)
- Indeterminaciones ($-\infty, \infty$): $-\text{Inf}$, Inf ($\frac{1}{0}$)

Operadores aritméticos

- Importancia de la jerarquía de operadores
- Operadores aritméticos
 - escalares
 - matriciales
- Operadores lógicos

Operadores aritméticos

<code>^</code>	potencia
<code>*</code> <code>/</code>	producto, cociente
<code>+</code> <code>-</code>	suma, resta
<code>%/%</code>	cociente entero
<code>%%</code>	módulo
<code>:</code>	generar una serie
<code>%*%</code>	producto matricial
<code>()</code>	paréntesis

Ejemplos

```
3 ^ 2
```

```
## [1] 9
```

```
3 ^ 1 + 1
```

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

```
3 ^ ( 1 + 1 )
```

```
## [1] 9
```

Ejemplos

```
10 / 2 * 5
```

```
## [1] 25
```

```
10 / 2 / 5
```

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

```
21 %% 5
```

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

Ejemplos

```
1:10
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
1:10 * 2
```

```
## [1] 2 4 6 8 10 12 14 16 18 20
```

```
2^(0:8)
```

```
## [1] 1 2 4 8 16 32 64 128 256
```

Operadores lógicos

!	no
== !=	igual, distinto
> >=	mayor, mayor o igual
< <=	menor, menor o igual
	o
& &&	y
#	comentario

Ejemplos

```
3 >=2
```

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

```
0 != 0.000000000000000001
```

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

```
5*2 > 9 & 3/2 == 1.5
```

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


Asignaciones

- `Variable <- expresión`
- Variable es un nombre que se utiliza como representación del resultado de una expresión

<code><-</code>	asignar a la izquierda
--------------------	------------------------

<code>-></code>	asignar a la derecha
--------------------	----------------------

<code>=</code>	asignar a la izquierda
----------------	------------------------

Ejemplos

```
a <- 3
```

```
a
```

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

```
a <- a + 1
```

```
a
```

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

```
(a <- a + 1)
```

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

rstudio: ¿qué objetos tengo y cuál es su valor?

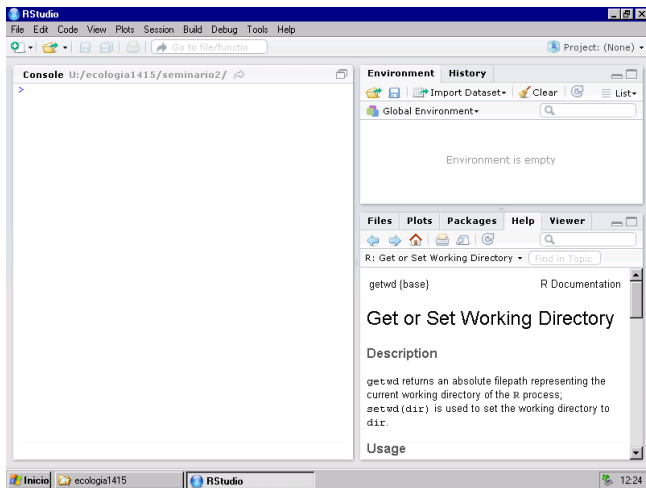


Figure 10:

rstudio: ¿qué objetos tengo y cuál es su valor?

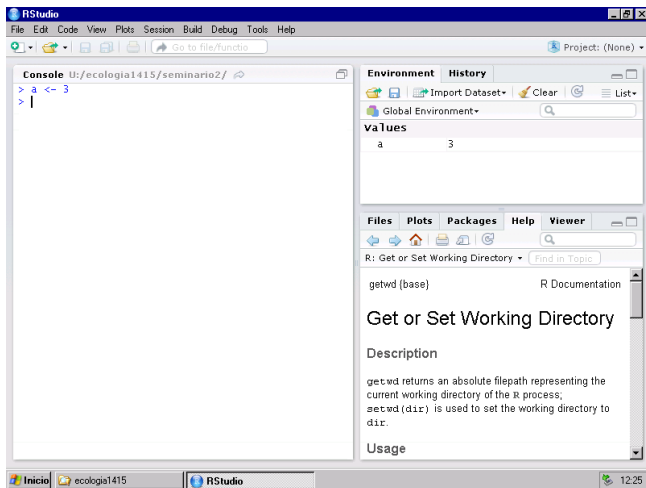


Figure 11:

Ejemplos

```
r <- 1  
area <- pi * r ^ 2  
longitud <- 2 * pi * r  
area
```

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

```
longitud
```

```
## [1] 6.283185
```

Ejemplos

```
r <- 1:10  
area <- pi * r ^ 2  
2 * pi * r -> longitud  
area ##;longitud
```

```
## [1] 3.141593 12.566371 28.274334 50.265482  
## [5] 78.539816 113.097336 153.938040 201.061930  
## [9] 254.469005 314.159265
```

Funciones

- Una función es un procedimiento para realizar una determinada tarea o cálculo
- función se asocia a un nombre, que sigue las mismas reglas que las variables
- **nombre_de_la_función** (*argumento 1, argumento 2, ...*)
- Los argumentos son propios de cada función
- En algunos casos los argumentos tienen valores por defecto

Ejemplo

```
log( 2 )
```

```
## [1] 0.6931472
```

```
log( 2, 10 )
```

```
## [1] 0.30103
```

```
log( exp( 1 ) )
```

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


Ejemplo

```
log( x = 2 , base = 10 )
```

```
## [1] 0.30103
```

```
log( base = 10 , x = 2 )
```

```
## [1] 0.30103
```

Funciones

`c()`

Concatenar los elementos que se indican, separados por comas

`seq()`

Generar una secuencia numérica

`rep()`

Generar un conjunto de valores repetidos

`sort()`

Ordena un vector

Funciones

<code>round()</code>	Redondeo de valores numéricos
<code>sqrt()</code>	Raíz cuadrada
<code>abs()</code>	Valor absoluto
<code>sin()</code>	Función trigonométricas seno
<code>cos()</code>	Función trigonométricas coseno
<code>log()</code>	Logaritmo natural
<code>exp()</code>	exponencial (e^x)

Funciones

<code>sum()</code>	Suma los elementos de un vector
<code>cumsum()</code>	Vector de sumas acumuladas
<code>max()</code>	Máximo de un vector
<code>min()</code>	Mínimo de un vector
<code>t()</code>	Transponer una matriz
<code>names()</code>	Nombres de filas o columnas
<code>nrow()</code>	Número de filas
<code>ncol()</code>	Número de columnas
<code>rownames()</code>	Nombre de las filas
<code>colnames()</code>	Nombres de las columnas

Funciones

<code>str()</code>	Proporciona información sobre la estructura de un objeto
<code>ls()</code>	Relación de objetos disponibles
<code>rm()</code>	Elimina uno o varios objetos
<code>read.table()</code>	Carga los datos de un fichero
<code>source()</code>	Carga el código de R escrito en un fichero

R: los objetos

Vectores

- Los vectores son un conjunto ordenado de valores
- Para calcular con todo el vector se emplea el nombre del objeto
- Para utilizar un subconjunto valores se emplea subíndices
- Los subíndices se incluyen entre corchetes ($x[3]$)
- Los subíndices están en el rango: 1 — *número de elementos del vector*
- Los subíndices pueden ser expresiones

Ejemplo

```
x <- c( 8, 5, 2, 4, 1, 6, 3 )  
length( x )
```

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

```
x
```

```
## [1] 8 5 2 4 1 6 3
```

```
x[]
```

```
## [1] 8 5 2 4 1 6 3
```


Ejemplo

```
x[ 1 ]
```

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

```
x[ 2:4 ]
```

```
## [1] 5 2 4
```

```
x[ c( 3, 5 ) ]
```

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

```
x[ -1 ]
```

```
## [1] 5 2 4 1 6 3
```

Matrices

- Una matriz es un conjunto ordenado de vectores
- Los elementos de la matriz están ordenados por filas y columnas
- Todos los vectores son del mismo tipo: enteros, caracteres, ...
- Los elementos de una matriz se identifican por dos subíndices
- El uso de los subíndices sigue las mismas reglas que en el caso de los vectores
- Se puede crear uniendo vectores o mediante la función `matrix()`

Ejemplo

```
m <- matrix( 1:12, 4, 3 )
```

```
m
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

```
m[ 1, ]
```

```
## [1] 1 5 9
```

Data frames

- Son semejantes a las matrices
- Se organizan por filas y columnas
- Las columnas no tienen por qué ser homogéneas
- Las columnas tienen nombre
- Habitualmente los *data frames* se obtienen de la lectura de un fichero de datos

Ejemplo: el fichero phlox.dat

```
read.table( "./files/ardilla.dat" )
```

	x	n	m
1	0	530	0.00
2	1	134	1.28
3	2	56	2.28
4	3	39	3.24
5	4	23	3.24
6	5	12	2.48
7	6	5	2.28
8	7	2	2.28

Ejemplo

```
phlox <- read.table(  
  "./files/phlox.dat" )  
head( phlox )
```

```
##      x    n semillas  
## 1     0 996         0  
## 2    63 668         0  
## 3   124 295         0  
## 4   184 190         0  
## 5   215 176         0  
## 6   264 172         0
```

Ejemplo

```
phlox[ , 3 ]
```

```
## [1] 0 0 0 0 0 0 0 53 485 803 973  
## [12] 95 0
```

```
phlox$n
```

```
## [1] 996 668 295 190 176 172 167 159 154 147 105  
## [12] 22 0
```

```
phlox[ , "n" ]
```

```
## [1] 996 668 295 190 176 172 167 159 154 147 105  
## [12] 22 0
```

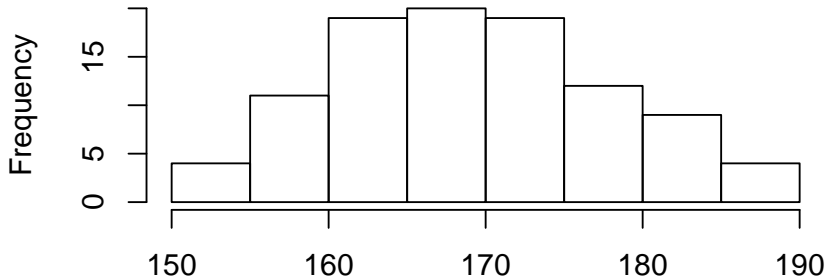
Listas

- Son objetos que pueden contener conjuntos heterogéneos de objetos
 - valores
 - vectores
 - matrices
 - *data frames*
 - listas
- Se suelen encontrar como resultado de funciones

Ejemplo

```
f <- "http://www.um.es/docencia/emc/datos/biom2003.dat"  
x <- read.table( f )  
hist( x$Altura ) -> xHist
```

Histogram of x\$Altura



Ejemplo

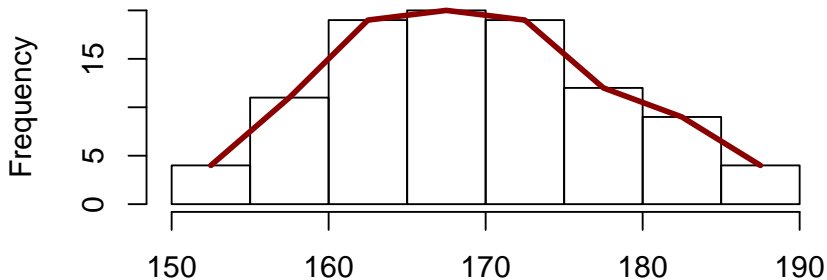
```
xHist
```

```
## $breaks
## [1] 150 155 160 165 170 175 180 185 190
##
## $counts
## [1] 4 11 19 20 19 12 9 4
##
## $density
## [1] 0.008163265 0.022448980 0.038775510
## [4] 0.040816327 0.038775510 0.024489796
## [7] 0.018367347 0.008163265
##
## $mids
## [1] 152.5 157.5 162.5 167.5 172.5 177.5 182.5
```

Ejemplo

```
plot( xHist, main = "Distribución estaturas" )  
lines( xHist$mids, xHist$count,  
       type="l", col="darkred", lwd = 3 )
```

Distribución estaturas



Trabajando con R

El desarrollo de los procedimientos

Preparación del área de trabajo

- Al iniciar la sesión de trabajo área de trabajo está vacía
- Primero deben cargarse las funciones necesarias
 - Mediante la función `source()`
 - Recurriendo a una librería
 - Recurriendo a un documento de *análisis reproducible*

Carga de datos

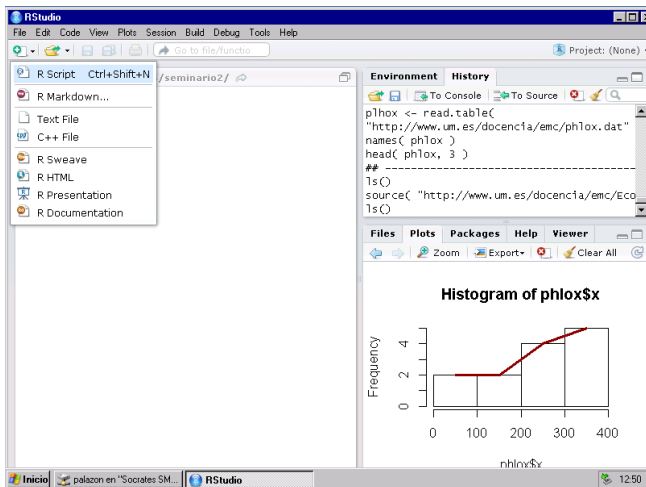
- Se cargan los datos a procesar asignándolos a las variables correspondientes.
- Se realizan los distintos cálculos y se copia el código utilizado en el block de notas o el editor favorito.
 - Se utiliza la función `savehistory("miSesion.R")`, desde la consola.
 - En `rstudio` se utiliza el icono del disquete en la pestaña de *History* para guardar.

Finalizar la sesión de trabajo

- Se cierra la sesión y se guarda la sesión y el fichero con el procedimiento, preferiblemente con la extensión `.R`

El histórico de la sesión

Creando un fichero de trabajo: script



The screenshot shows the RStudio interface. The main editor window displays an R script with the following code:

```
plhox <- read.table(  
  "http://www.um.es/docencia/emc/phlox.dat"  
  names( phlox )  
  head( phlox, 3 )  
  ## -----  
  ls()  
  source( "http://www.um.es/docencia/emc/Eco  
  ls()
```

The Environment pane on the right shows the variable `plhox` as a data frame. Below the code editor, the Plots pane displays a histogram titled "Histogram of plhox\$x". The x-axis is labeled "plhox\$x" and ranges from 0 to 400. The y-axis is labeled "Frequency" and ranges from 0 to 4. The histogram shows four bars with frequencies of approximately 2, 2, 4, and 4. A red line is overlaid on the histogram, representing a normal distribution fit.

Figure 12:

Copiando el histórico

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, and Help. The main window is divided into several panes:

- Source Editor:** Contains a script with the following code:

```
1 |  
r <- 1:10  
area <- pi * r ^ 2  
2 * pi * r -> longitud  
area #; longitud
```
- Console:** Shows the execution of the code and the resulting data frame:

```
U:/ecologia1415/seminario2/ <= >  
> r <- 1:10  
> area <- pi * r ^ 2  
> 2 * pi * r -> longitud  
> area #; longitud  
[1] 3.141593 12.566371 28.274334 50.265482  
[5] 78.539816 113.097336 153.938040 201.061930  
[9] 254.469005 314.159265  
> |
```
- Environment/History:** Shows the current environment with variables: longitud, ##, r, area, 2 * pi * r -> longitud, area #; longitud, ##, and longitud. A blue highlight is visible over the code in the source editor.
- Plots:** Displays a histogram titled "Histogram of phlox\$x". The x-axis is labeled "phlox\$x" and ranges from 0 to 400. The y-axis is labeled "Frequency" and ranges from 0 to 4. The histogram shows a distribution with a peak frequency of 4.

The taskbar at the bottom shows the system tray with the time 13:14 and the active window U:/ecologia1415/seminari...

Figure 13:

Un script para reutilizar

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains an R script with the following code:

```
1 r <- 1:10
2 area <- pi * r ^ 2
3 2 * pi * r -> longitud
4 area #;longitud
5
```
- Environment/History:** Shows the execution history:

```
longitud
##-----##
r <- 1:10
area <- pi * r ^ 2
2 * pi * r -> longitud
area #;longitud
##-----##
longitud
```
- Console:** Shows the output of the script execution:

```
U:/ecologia1415/seminario2/ >
> r <- 1:10
> area <- pi * r ^ 2
> 2 * pi * r -> longitud
> area #;longitud
[1] 3.141593 12.566371 28.274334 50.265482
[5] 78.539816 113.097336 153.938040 201.061930
[9] 254.469005 314.159265
>
```
- Plots:** Displays a histogram titled "Histogram of phlox\$x". The x-axis is labeled "phlox\$x" and ranges from 0 to 400. The y-axis is labeled "Frequency" and ranges from 0 to 4. The histogram shows a distribution with a peak frequency of 4 at the highest bin.

Figure 14:

RR, más allá: reproducible research

Un fichero Rmd: mezcla de texto y R

The screenshot shows the RStudio interface with an R Markdown file open. The console displays the execution of R code, and the Plots pane shows a histogram of the variable 'phlox\$x'.

```
Environment History
longitud
##-----##
r <- 1:10
area <- pi * r ^ 2
2 * pi * r -> longitud
area #;longitud
##-----##
log( 2 )
```

```
Files Plots Packages Help Viewer
Zoom Export Clear All
```

Histogram of phlox\$x

Frequency

0 2 4

0 100 200 300 400

phlox\$x

```
U:/ecologia1415/seminario2/
> r <- 1:10
> area <- pi * r ^ 2
> 2 * pi * r -> longitud
> area #;longitud
[1] 3.141593 12.566371 28.274334 50.265482
[5] 78.539816 113.097336 153.938040 201.061930
[9] 254.469005 314.159265
>
```

Figure 15:

Rmd: algo más que un *script*

The screenshot shows the RStudio interface with an R Markdown document open. The document is titled 'ejemploMd.Rmd' and contains the following code:

```
4 ---
5 # Dinámica poblacional de 'phlox'
6
7 ## Carga de datos
8
9 ```{r}
10 phlox <- read.table(
11 "http://fobos.inf.um.es/R/ecologia/
12 .dat" )
13
14 ## visualizando valores
15 ```{r}
16 head( phlox, 2 )
17
18
19 ## un gráfico
20 ```{r}
21 hist( phlox$x ) -> xHist
22
23 + (Top Level) +
```

The rendered HTML output is shown in the right pane, titled 'ejemploMd.html'. It contains the following sections:

Carga de datos

```
phlox <- read.table(
"http://fobos.inf.um.es/R/ecologia/phlox.dat" )
```

Visualizando valores

```
head( phlox, 2 )
```

##	x	n	semillas
## 1	0	996	0
## 2	63	668	0

Un gráfico

```
hist( phlox$x ) -> xHist
```

Histogram of phlox\$x

The console at the bottom shows the command prompt and the output of the histogram function.

Figure 16:

Preguntas

¿Cómo seguir avanzando con R?

Cursos de R

- Básico, para los interesados: Try R, curso interactivo *on line* breve y muy práctico.
- Cursos *on line* de las distintas plataformas: Miriada X, Coursera, edX, . . .
- *Open Course Ware* (OCW), busca "read.table"
- CRAN: *Contributed Documentation*
- Libros
- . . .

¿Más preguntas?