



Instalación y uso básico de RStudio

Pedro A. Castillo Valdivieso
Universidad de Granada

pacv@ugr.es





Objetivos

- Descargar e instalar RStudio
- Aprender a trabajar con Rstudio (y con R)
- Hacer operaciones básicas con datos
- Usar estructuras de datos
- Realizar operaciones estadísticas
- Representar gráficamente conjuntos de datos



Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?





¿Qué es R?

Software para hacer análisis estadístico de datos.

Lenguaje muy utilizado en investigación en biomedicina, estadística, bioinformática y matemáticas financieras.

Disponible para casi cualquier sistema operativo:
Windows, Linux, OS-X.

Software libre.

¿Qué es R?

R proporciona diversas herramientas estadísticas y gráficas.

[https://en.wikipedia.org/wiki/R_\(programming_language\)](https://en.wikipedia.org/wiki/R_(programming_language))

<http://cran.r-project.org>





Descargar R

Descargar de:

<http://www.r-project.org>

<http://cran.es.r-project.org>

<https://cran.rediris.es/>

- Instalación sencilla.
- Seleccionar todas las opciones para disponer de toda la ayuda.



Ejecutar R

Tras instalarlo, se puede ejecutar con el acceso directo en el escritorio:



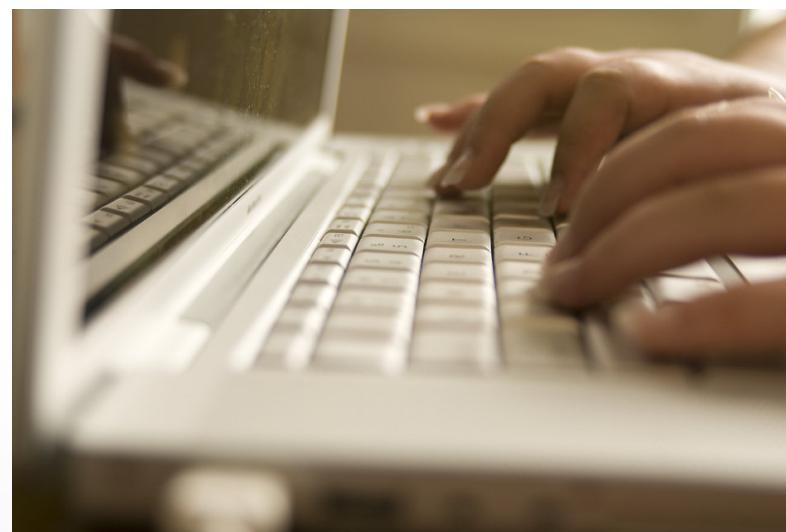
La interfaz básica de R es una consola con el prompt (el símbolo ">") donde ejecutar las expresiones en R.

Ejecutar R

R no dispone de un menú con todas las herramientas o técnicas estadísticas disponibles en el lenguaje.

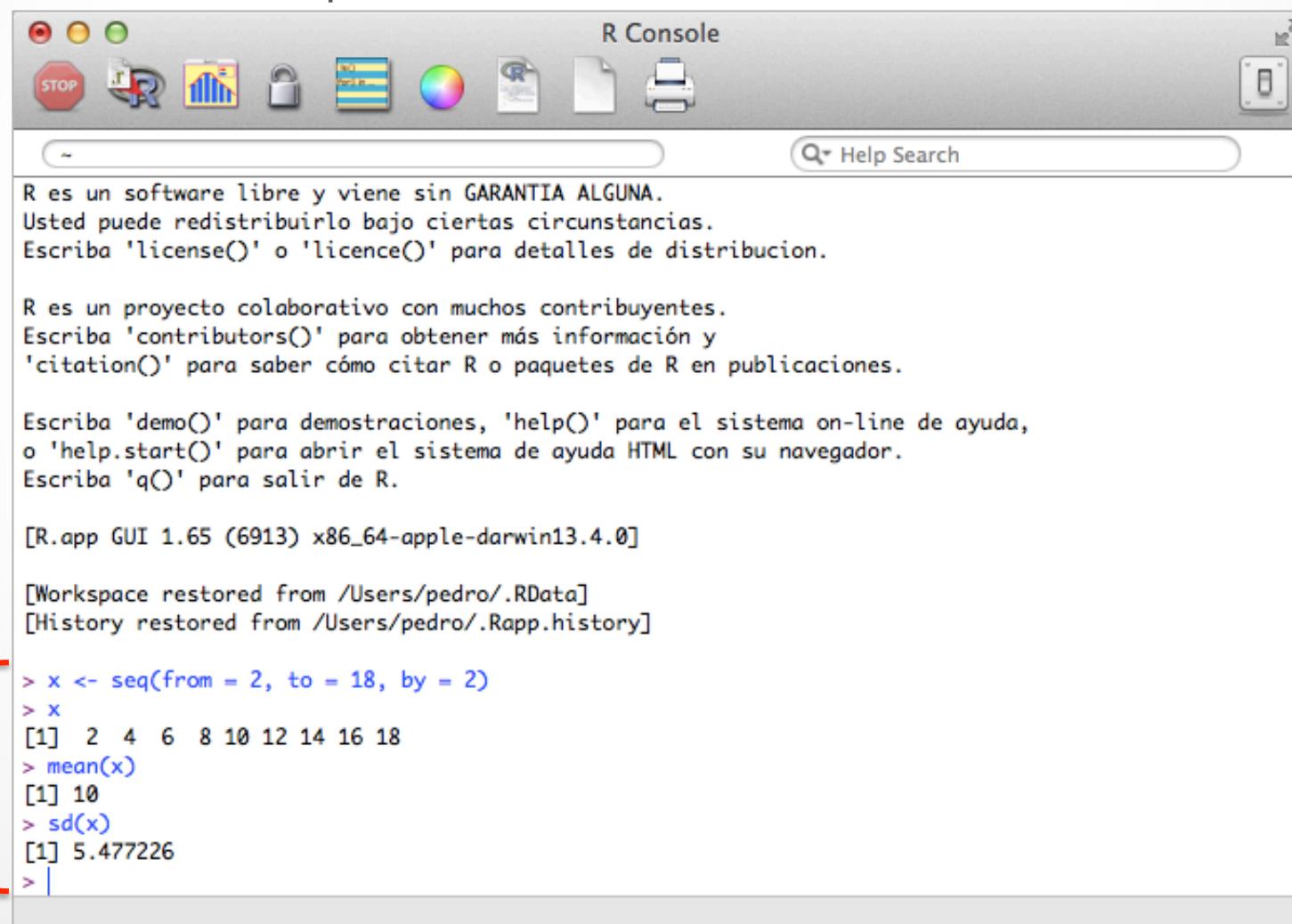
Para realizar una operación en R debemos teclear la expresión en la consola...

O bien ejecutar un conjunto de comandos escritos en un fichero.



Ejecutar R

Disponemos de una barra de herramientas para acceder a las operaciones frecuentes:



R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

[R.app GUI 1.65 (6913) x86_64-apple-darwin13.4.0]

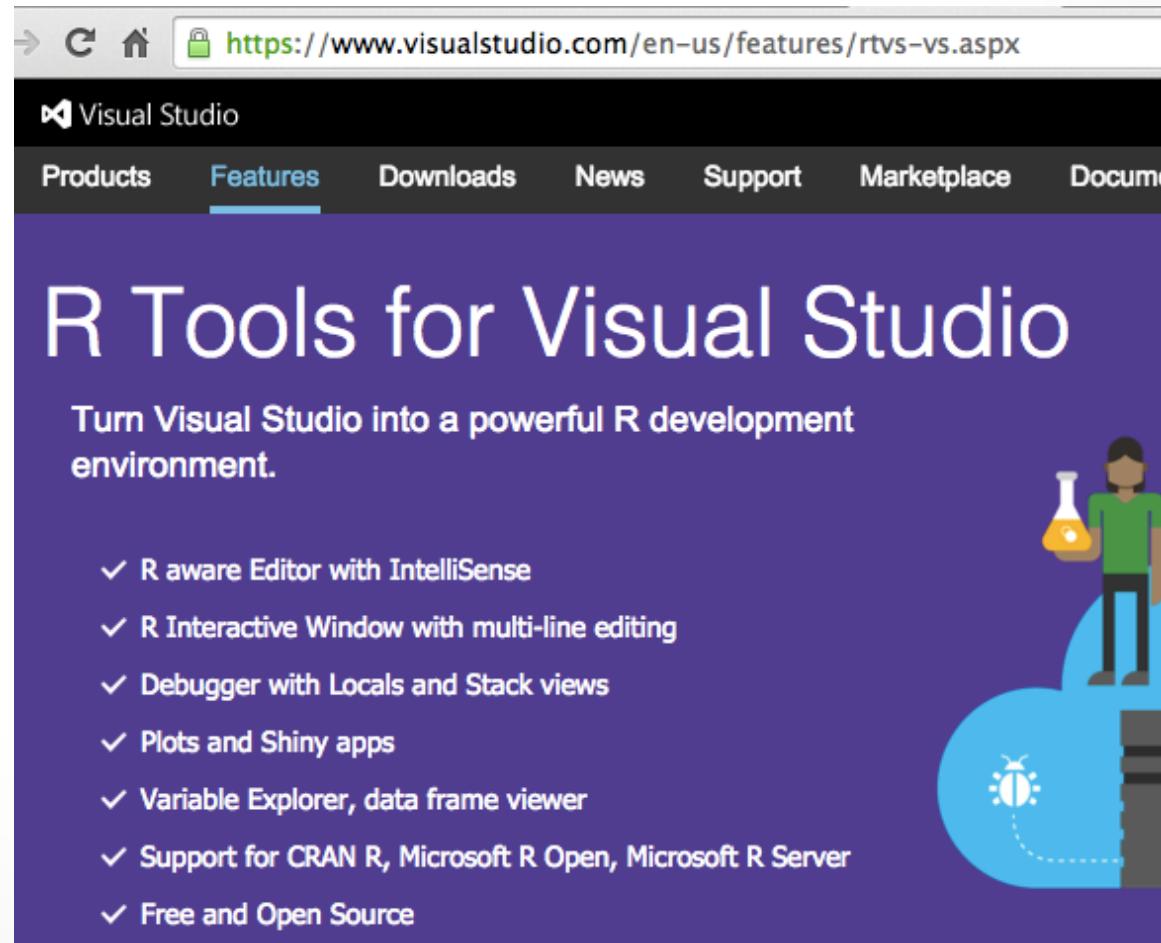
[Workspace restored from /Users/pedro/.RData]
[History restored from /Users/pedro/.Rapp.history]

```
> x <- seq(from = 2, to = 18, by = 2)
> x
[1] 2 4 6 8 10 12 14 16 18
> mean(x)
[1] 10
> sd(x)
[1] 5.477226
>
```

Ejecutar R

Sin embargo, disponemos de interfaces que facilitan el trabajo con R:

- RStudio
- Visual Studio
(Microsoft)



The screenshot shows a web browser displaying the Visual Studio website at <https://www.visualstudio.com/en-us/features/rts-vs.aspx>. The page is titled "R Tools for Visual Studio" and features a sub-headline: "Turn Visual Studio into a powerful R development environment." To the right, there's an illustration of a scientist holding a flask. A vertical sidebar on the right lists several features of the tool.

R Tools for Visual Studio

Turn Visual Studio into a powerful R development environment.

- ✓ R aware Editor with IntelliSense
- ✓ R Interactive Window with multi-line editing
- ✓ Debugger with Locals and Stack views
- ✓ Plots and Shiny apps
- ✓ Variable Explorer, data frame viewer
- ✓ Support for CRAN R, Microsoft R Open, Microsoft R Server
- ✓ Free and Open Source



Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?





¿Qué es RStudio?

RStudio es una interfaz que facilita el uso de R.

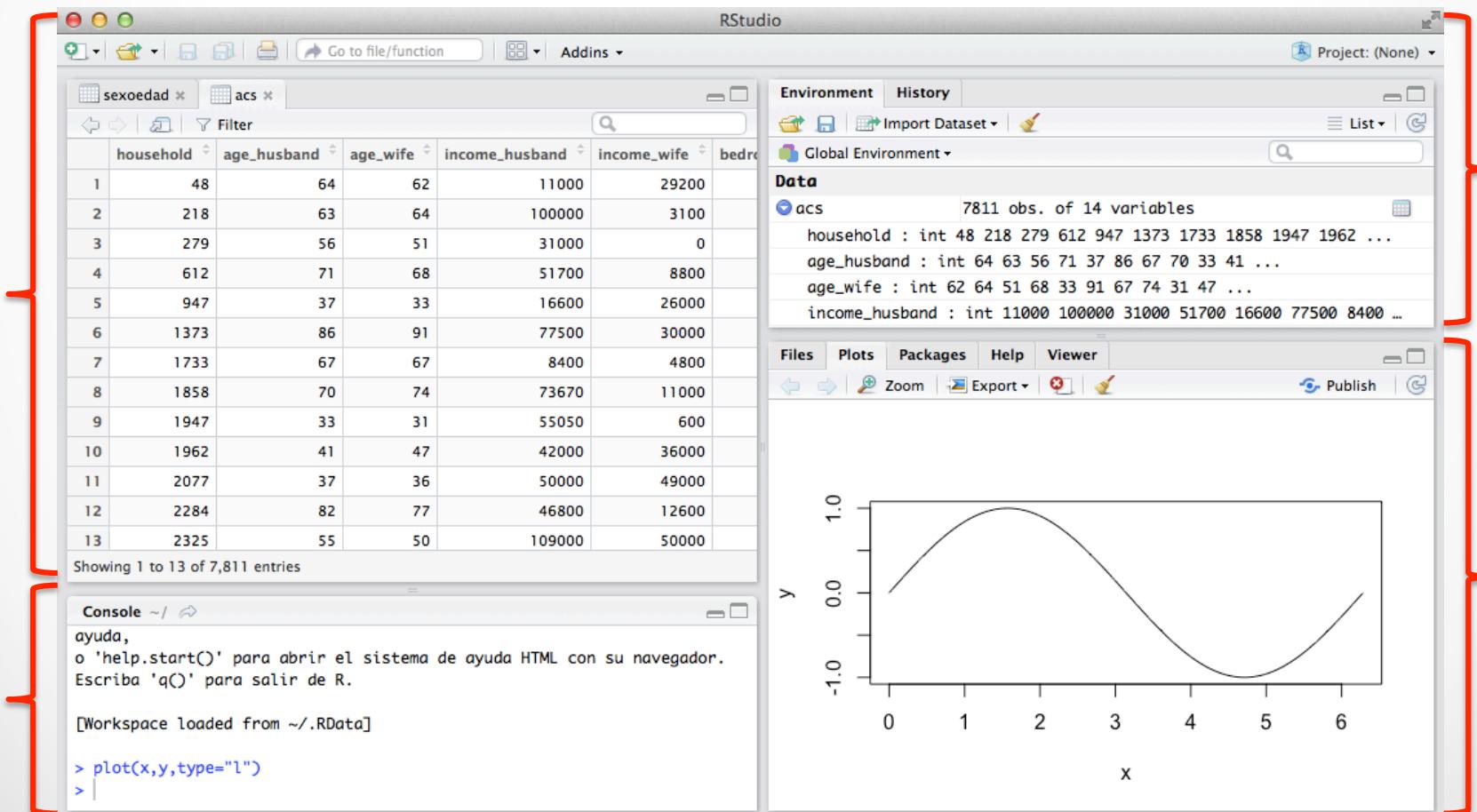
RStudio necesita tener instalado R previamente.

<https://es.wikipedia.org/wiki/RStudio>

Esta interfaz incluye una consola, un editor para facilitar la ejecución de código, y herramientas para la depuración y la gestión del espacio de trabajo.

Objetivo de RStudio

- Dar soporte a los procedimientos y las técnicas de análisis de R.
- Proporcionar un entorno amigable.



R en Visual Studio (MS)

- Visual Studio (Microsoft)

rproject13 - Microsoft Visual Studio

File Edit View Project Debug Team Tools R Tools Window Help

Quick Launch (Ctrl+Q) sean mortaz SM

Variable Explorer

Global Environment

Name	Value	Class	Type
mtcars	32 obs. of 11 variables	data.frame	list
@.Data	List of 11	list	list
@names	chr [1:11] "mpg" "cyl" "disp" "hp" "drat" "wt"	character	character
@row.names	chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" "Hornet Sportabout" "Valiant" "Duster 360" "Merc 240D" "Merc 230" "Merc 280" "Merc 280C" "Merc 450SE"	character	character
[1]	"Mazda RX4"	character	character
[2]	"Mazda RX4 Wag"	character	character
[3]	"Datsun 710"	character	character
[4]	"Hornet 4 Drive"	character	character
[5]	"Hornet Sportabout"	character	character
[6]	"Valiant"	character	character
[7]	"Duster 360"	character	character
[8]	"Merc 240D"	character	character

R Data: base::GlobalEnv\$mtcars

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6cyl	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6cyl	160.0	110	3.90	2.875
Datsun 710	22.8	4cyl	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6cyl	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8cyl	360.0	175	3.15	3.440
Valiant	18.1	6cyl	225.0	105	2.76	3.460
Duster 360	14.3	8cyl	360.0	245	3.21	3.570
Merc 240D	24.4	4cyl	146.7	62	3.69	3.190
Merc 230	22.8	4cyl	140.8	95	3.92	3.150
Merc 280	19.2	6cyl	167.6	123	3.92	3.440
Merc 280C	17.8	6cyl	167.6	123	3.92	3.440
Merc 450SE	16.4	8cyl	275.8	180	3.07	4.070

R interactive

```
> library(ggplot2)
> # create factors with value labels
> mtcars$gear <- factor(mtcars$gear, levels = c(3, 4, 5),
+   labels = c("3gears", "4gears", "5gears"))
> mtcars$am <- factor(mtcars$am, levels = c(0, 1),
+   labels = c("Automatic", "Manual"))
> mtcars$cyl <- factor(mtcars$cyl, levels = c(4, 6, 8),
+   labels = c("4cyl", "6cyl", "8cyl"))
> # Kernel density plots for mpg
> # grouped by number of gears (indicated by color)
> qplot(mpg, data = mtcars, geom = "density", fill = gear, alpha = I(.5),
+   main = "Distribution of Gas Mileage", xlab = "Miles Per Gallon",
+   ylab = "Density")
> # Scatterplot of mpg vs. hp for each combination of gears and cylinders
```

R Plot

Distribution of Gas Mileage

gear

14

¿Por qué utilizar RStudio?

- IDE diseñado para el lenguaje R.
- Multiplataforma (Windows, Linux, Mac).
- Ayuda integrada.
- Gestión de paquetes.
- Gestionar la sesión (objetos, historial de comandos).
- Varios scripts abiertos a la vez.
- Depuración de código.
- Autocompletado de código.
- Presentaciones en HTML5 o Sweave.
- Trabajar con markdown o knitr.



Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?



Instalar RStudio

La instalación de RStudio se realiza descargando el paquete correspondiente:

<http://www.rstudio.org>

<https://www.rstudio.com/products/rstudio/download/#download>

The screenshot shows the official RStudio website at <https://www.rstudio.com>. The page has a blue header with the URL. The main content area features a large white "R" logo on a blue circular background. The text "Welcome to RStudio - Open source and enterprise-ready professional software for R" is displayed. At the bottom, there are three buttons: "Download RStudio", "Discover Shiny", and "shinyapps.io Login". Below the buttons is a navigation bar with five dots, where the first two are white and the others are grey.

Descargar RStudio

 <https://www.rstudio.com/products/rstudio-desktop/>



Download RStudio Desktop

RStudio Desktop lets you access RStudio locally on your Windows, Mac, or Linux computer, giving you access to a set of integrated tools designed to help you be more productive with R.

RStudio Server Pro delivers additional benefits including priority support and a commercial license for organizations not able to use AGPL software.

To run RStudio on a server and access it from any web browser, please download RStudio Server (available in Open Source in Pro editions).

DOWNLOAD RSTUDIO DESKTOP OPEN SOURCE



Descargar RStudio

Disponible para cualquier sistema:

The screenshot shows a web browser displaying the RStudio download page at <https://www.rstudio.com/products/rstudio/download/#download>. The page header includes the RStudio logo, navigation links for Products, Resources, Pricing, About Us, and Help, and social sharing icons. A note at the top states: "The 32 bit version of R, you can use an older version of RStudio." Below this, a section titled "Installers for Supported Platforms" contains a table with the following data:

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - Mac OS X 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46cfb0aea75
RStudio 1.2.1335 - Fedora 19+/RedHat 7+ (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a

Instalar RStudio Desktop

La instalación de RStudio es sencilla y guiada.

Sólo hay que seguir las instrucciones que ofrece el instalador.

Tras instalarlo, se puede ejecutar con el acceso directo en el escritorio:





Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?





Usando RStudio

La interfaz de RStudio está dividida en 4 partes:

A la izquierda abajo, donde está el prompt “>” es la consola o espacio de trabajo.

A la izquierda arriba está el editor de texto (scripts R).

A la derecha arriba está el historial de objetos almacenados en memoria.

A la derecha abajo está el directorio de trabajo, los gráficos que se van generando, paquetes y la ayuda.



Usando RStudio

Editor de scripts en R

Objetos en memoria

The screenshot shows the RStudio interface with several panels:

- Editor de scripts en R:** The top-left panel displays an R script named "Untitled1". The code shown is:

```
1
2 help(mean)
3
4 k <- c(7, 4, 5, 9, 1)
5 sort(x)
```
- Objetos en memoria:** The top-right panel, titled "Environment", lists objects in the Global Environment:

Data	Description
acs	7811 obs. of 14 variables
sexoedad	10 obs. of 4 variables
- Consola (ejecución de órdenes):** The bottom-left panel, titled "Console", shows the output of the R script:

```
[Workspace loaded from ~/.RData]

> help(mean)
> x <- c(7, 4, 5, 9, 1)
> sort(x)
[1] 1 4 5 7 9
>
```
- Directorio de trabajo, gráficos, ayuda:** The bottom-right panel, titled "Help", provides documentation for the "mean" function:

mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

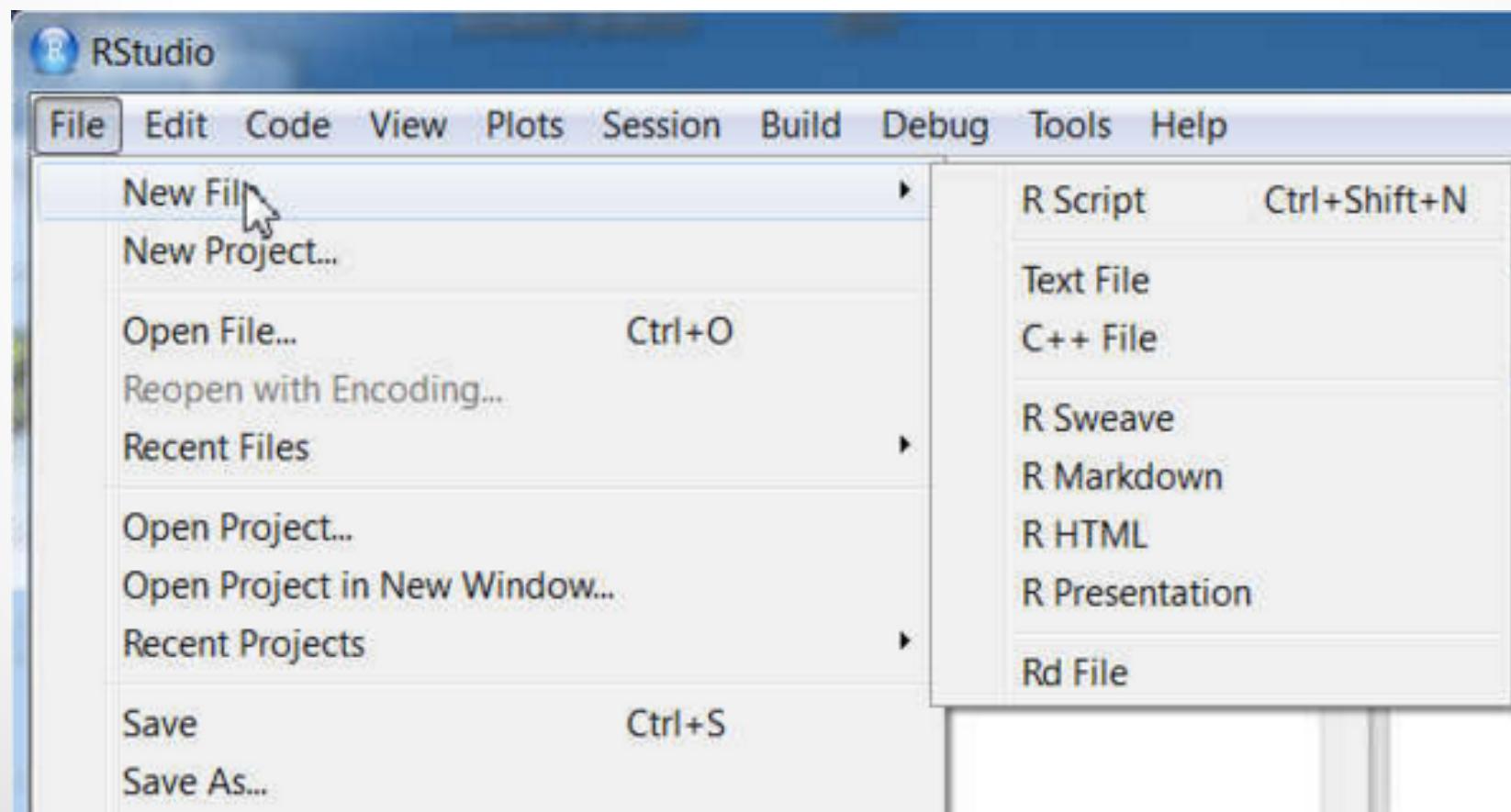
Default S3 method:

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

Usando RStudio

En RStudio tenemos un Menú principal con el cual, de forma sencilla, podemos modificar los datos, aplicar funciones estadísticas, numéricas, o generar gráficos.





Usando RStudio

Con File->New File->R Script podemos abrir la ventana de Edición para tener un entorno donde manipular el código R.

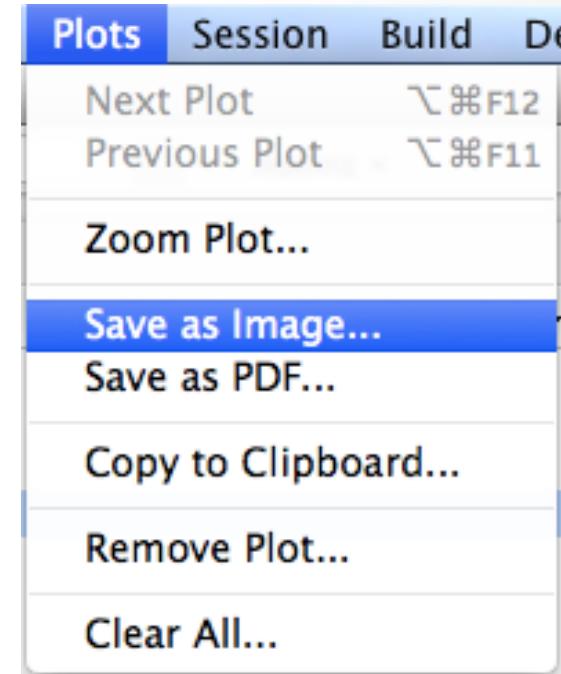
En esta ventana escribimos las instrucciones y para ejecutarlas pulsamos Ctrl+Intro

Con Edit->Clear Console despejamos la consola de todos los resultados parciales que tengamos en pantalla. No borra los objetos almacenados anteriormente en la memoria.

Usando RStudio

El menú PLOTS permite gestionar los gráficos que hacemos:

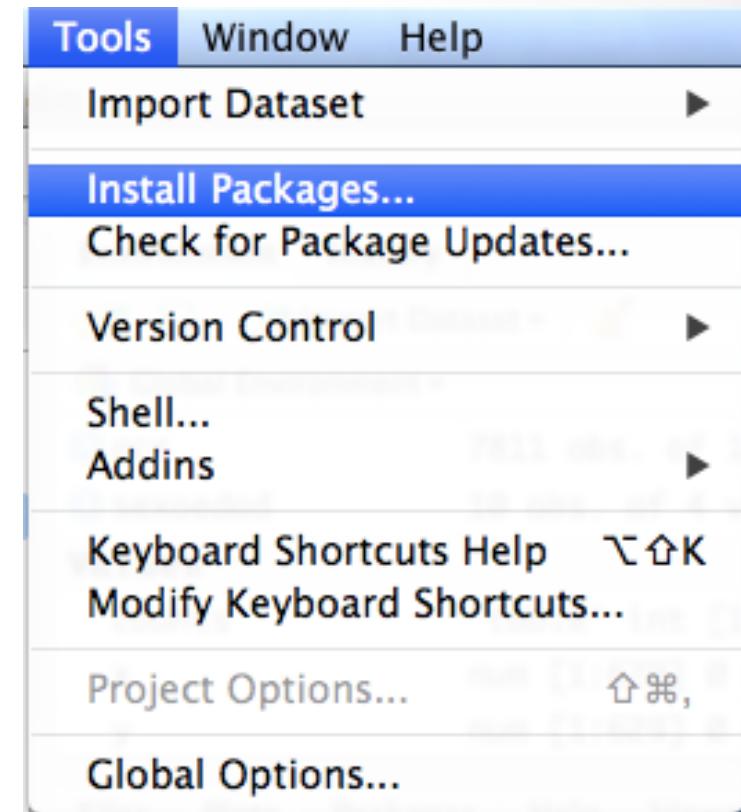
- Next Plot
- Previous Plot
- Save Plot as Image
- Save Plot as PDF
- Copy Plot to Clipboard
- Remove Plot
- Clear All



Usando RStudio

El menú TOOLS permite gestionar y configurar el entorno:

- Install Packages
- Check for Package Updates





Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?





El lenguaje R

Usar scripts de R para agrupar varias órdenes.

Útil para ejecutar un mismo código varias veces.

Todo se guarda en la sesión de trabajo (variables, funciones, datos, órdenes introducidas, etc).

Guardar la sesión para recuperarla más tarde y seguir trabajando justo donde estábamos (opciones “Archivo/Cargar área de trabajo” y “Archivo/Guardar área de trabajo”)

El lenguaje R

Obtener ayuda sobre cualquier función:

`help(mean)`

`?mean`

A continuación veremos:

- Tipos de datos
- Cálculos
- Estructuras de datos
- Manejo de vectores y matrices
- Generar secuencias
- Leer datos de archivos
- El espacio de trabajo

R. Tipos de datos

R soporta distintos tipos de datos: numeric, integer, double, logical, character

```
real <- 4  
real  
  
entero = as.integer(4)  
entero  
  
cadena = "Hola"  
cadena
```





R. Tipos de datos

The screenshot shows the RStudio interface with several panels:

- Script Editor (Top Left):** Displays the code in `ejemplos_presentacion.R`. A red bracket on the left side groups the first 13 lines of code.
- Environment (Top Right):** Shows the global environment with objects like `acs`, `sexedad`, `cadena`, `counts`, and `entero`.
- File Browser (Bottom Right):** Shows the contents of the `ejemplos` directory, including files like `acs_or.csv`, `ANOVA`, and `ejRnw`.
- Console (Bottom Left):** Displays the output of running the R code from the script editor.

```
ejemplos_presentacion.R x
1 # ----- transp. 32
2 real <- 4
3 real
4
5 entero = as.integer(4)
6 entero
7
8 cadena = "Hola"
9 cadena
10
11 # ----- transp.
12
13 # ----- transp.

[1] (Top Level)  R Script

[Workspace loaded from ~/.RData]

> real <- 4
> real
[1] 4
>
> entero = as.integer(4)
> entero
[1] 4
>
> cadena = "Hola"
> cadena
[1] "Hola"
>
```

Output from the Console:

```
[Workspace loaded from ~/.RData]

> real <- 4
> real
[1] 4
>
> entero = as.integer(4)
> entero
[1] 4
>
> cadena = "Hola"
> cadena
[1] "Hola"
>
```

R. Cálculos

R tiene implementadas las principales operaciones aritméticas: +, -, *, /, ^, %/ y %%

y otras funciones muy utilizadas:

- `sqrt(x)` Raíz cuadrada de x
- `abs(x)` Valor absoluto de x
- `sin(x)` Seno de x
- `cos(x)` Coseno de x
- `tan(x)` Tangente de x
- `log(x)` Logaritmo neperiano de x
- `log10(x)` Logaritmo decimal de x
- `exp(x)` Exponencial de x

R. Tipos de datos

Ejercicios:

- (1) Definir tres variables (x , y , z) con los valores 7, 8 y 3 respectivamente. Calcular sobre otra variable la expresión $w=x*y+z$
- (2) Calcular el seno de pi.
- (3) Calcular la raíz cuadrada de 9.
¿Qué ocurre si calculamos la raíz cuadrada de -1?
¿Y si se la calculamos al número complejo $3+5i$?

R. Estructuras de datos

Vectores: Una dimensión.

Matrices: Dos dimensiones.

Arrays: Tres o más dimensiones.

Data-frames: almacena variables de distintos tipos, pero con la misma longitud.

Listas: almacena variables de cualquier tipo y longitud.

R. Vectores

Una dimensión.

Se definen con la función c().

Se indexan con []

```
x <- c(5, 3, 8, 2, 9)
```

```
x
```

```
x[3]
```





R. Vectores

RStudio

ejemplos_presentacion.R x

Source on Save | Go to file/function | Addins | Project: (None)

```
6 entero
7
8 cadena = "Hola"
9 cadena
10
11 # ----- transp. 36
12 x <- c(5, 3, 8, 2, 9)
13
14 x
15
16 x[3]
17
18 # ----- transp.
```

12:1 (Top Level) R Script

Console ~ /

```
> entero
[1] 4
>
> cadena = "Hola"
> cadena
[1] "Hola"
> x <- c(5, 3, 8, 2, 9)
>
> x
[1] 5 3 8 2 9
>
> x[3]
[1] 8
>
> |
```

Environment | History

Import Dataset | Global Environment | List | G

sexedad 10 obs. of 4 variables

Variables	Type	Value
cadena	character	"Hola"
counts	table	int [1:7(1d)] 24 198 1475 4040 ...
entero	integer	4L
real	double	4
x	vector	num [1:5] 5 3 8 2 9

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More | G

Home > ejemplos

- acs_or.csv 721.5 KB Feb 15, 2016, 11:35 PM
- ANOVA
- ejemplos_presentacion.R 404 B Mar 5, 2016, 11:12 PM
- ejRnw
- example01.txt 230 B Feb 12, 2016, 10:21 AM
- funcmat.csv 716 B Feb 12, 2016, 12:33 PM
- funcmat.txt 717 B Feb 12, 2016, 12:35 PM
- funcmat.xlsx 44.6 KB Feb 12, 2016, 11:14 AM
- K-W
- misfunciones.R 148 B Mar 4, 2016, 3:21 PM
- ordenes.txt 4.1 KB Mar 4, 2016, 3:21 PM
- sexedad.txt 142 B Feb 14, 2016, 12:59 AM

37

R. Vectores

La concatenación de vectores se puede hacer de varias formas:

```
# concatenar dos vectores  
  
b <- 1  
b <- c(b, 2)  
  
v1 <- sample(1:10)  
v2 <- sample(11:20)  
v3 <- c(v1,v2)
```





R. Vectores

RStudio

ejemplos_presentacion.R*

Source on Save | Go to file/function | Addins | Project: (None)

```
15
16 x[3]
17
18 # ----- transp. 38
19 # concatenar dos vectores
20
21 b <- 1
22 b <- c(b, 2)
23
24 v1 <- sample(1:10)
25 v2 <- sample(11:20)
26 v3 <- c(v1,v2)
27
```

27:1 (Top Level) R Script

Console ~/

```
>
> b <- 1
> b <- c(b, 2)
> b
[1] 1 2
> v1 <- sample(1:10)
> v2 <- sample(11:20)
> v3 <- c(v1,v2)
>
> v3
[1] 3 7 8 10 5 4 1 2 6 9 12 16 13 14 18 19 15 11
[19] 17 20
>
>
> |
```

Environment History

Import Dataset

Global Environment

values	
b	num [1:2] 1 2
cadena	"Hola"
counts	'table' int [1:7(1d)] 24 198 1475 4040 ...
entero	4L
real	4
v1	int [1:10] 3 7 8 10 5 4 1 2 6 9
v2	int [1:10] 12 16 13 14 18 19 15 11 17 20

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home > ejemplos

acs_or.csv	721.5 KB	Feb 15, 2016, 11:35 PM	
ANOVA			
ejemplos_presentacion.R	404 B	Mar 5, 2016, 11:12 PM	
ejRnw			
example01.txt	230 B	Feb 12, 2016, 10:21 AM	
funcmat.csv	716 B	Feb 12, 2016, 12:33 PM	
funcmat.txt	717 B	Feb 12, 2016, 12:35 PM	
funcmat.xlsx	44.6 KB	Feb 12, 2016, 11:14 AM	
K-W			
misfunciones.R	148 B	Mar 4, 2016, 3:21 PM	
ordenes.txt	4.1 KB	Mar 4, 2016, 3:21 PM	
sexoedad.txt	142 B	Feb 14, 2016, 12:59 AM	

39

R. Ordenar vectores

- rank(x)** da el orden en el que están los números
- order(x)** da los índices del vector, ordenados
- sort(x)** ordena el vector
- rev(x)** invierte los elementos (sin ordenar)

```
a <- c(45, 50, 10, 96)
order(a)
rank(a)
sort(a)
rev(a)
sort(a, decreasing=TRUE)
```



R. Ordenar vectores

RStudio

ejemplos_presentacion.R x

Source on Save | Import Dataset | Addins | Project: (None)

```
60 # ----- transp. 46
61 a <- c(45, 50, 10, 96)
62 order(a)
63 rank(a)
64 sort(a)
65 rev(a)
66 sort(a, decreasing=TRUE)
67 # ----- transp.
68
69 # ----- transp.
```

Environment | History

Global Environment

dataframe	3 obs. of 3 variables
m	num [1:2, 1:3] 1 2 3 4 5 6
m2	num [1:2, 1:3] 1 4 2 5 3 6
midataframe	5 obs. of 7 variables
sexoedad	10 obs. of 4 variables

Values

a	num [1:4] 45 50 10 96
---	-----------------------

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

Home > ejemplos

- ..
- acs_or.csv 721.5 KB Feb 15, 2016, 11:35 PM
- ANOVA
- ejemplos_presentacion.R 1.2 KB Mar 5, 2016, 11:52 PM
- ejRnw
- example01.txt 230 B Feb 12, 2016, 10:21 AM
- funcmat.csv 716 B Feb 12, 2016, 12:33 PM
- funcmat.txt 717 B Feb 12, 2016, 12:35 PM
- funcmat.xlsx 44.6 KB Feb 12, 2016, 11:14 AM
- K-W

Console ~ /

```
> a <- c(45, 50, 10, 96)
> order(a)
[1] 3 1 2 4
> rank(a)
[1] 2 3 1 4
> sort(a)
[1] 10 45 50 96
> rev(a)
[1] 96 10 50 45
> sort(a, decreasing=TRUE)
[1] 96 50 45 10
>
>
```

R. Secuencias

Hay diversas formas de generar secuencias de números (muy útiles en diversas situaciones):

```
x <- c(1, 2, 3, 4, 5)
x[3]

x <- 1 : 10
x

x <- seq(from = 2, to = 18, by = 2)
x[3]
```

R. Matrices

Dos dimensiones.

Se definen con la función matrix().

```
m <- matrix(c(1,2,3,4,5,6), nrow = 2, ncol = 3)
m
[,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
m[2, 3]
[1] 6
m[1, ]
[1] 1 3 5
```



R. Matrices

Dos dimensiones.

Se definen con la función matrix().

```
m2 <- matrix(c(1,2,3,4,5,6),  
              nrow = 2, ncol = 3,  
              byrow = TRUE)  
  
m2  
[,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6
```



R. Data-frames

Almacenan variables de distintos tipos, pero con la misma longitud.

Se definen con la función `data.frame()`.

```
a <- c(11,12,13)
b <- c("uno", "dos", "tres")
c <- c(TRUE, FALSE, TRUE)

dataframe <- data.frame(a, b, c)
dataframe
  a   b   c
1 11 uno TRUE
2 12 dos FALSE
3 13 tres TRUE

dataframe$a
[1] 11 12 13

dataframe$a[1]
[1] 11
```



R. Data-frames

Almacenan variables de distintos tipos, pero con la misma longitud.

Se definen con la función `data.frame()`.

```
midataframe = read.table(header=T, text="  
name    age     hgt     wgt     race   year   SAT  
Bob     21      70      180     Cauc   Jr     1080  
Fred    18      67      156     Af.Am  Fr     1210  
Barb    18      64      128     Af.Am  Fr     840  
Sue     24      66      118     Cauc   Sr     1340  
Jeff    20      72      202     Asian   So     880  
")
```

```
midataframe$age  
midataframe[1,5]  
hist(midataframe$age)
```





R. Data-frames

RStudio

ejemplos_presentacion.R*

Source on Save | Go to file/function | Addins | Project: (None)

```
50
51 # ----- transp. 43
52 midataframe = read.table(header=T, text="
53 name age hgt wgt race year SAT
54 Bob 21 70 180 Cauc Jr 1080
55 Fred 18 67 156 Af.Am Fr 1210
56 Barb 18 64 128 Af.Am Fr 840
57 Sue 24 66 118 Cauc Sr 1340
58 Jeff 20 72 202 Asian So 880
59 ")
60
```

(Top Level) R Script

Console ~ /

```
+ ")
> midataframe
  name age hgt wgt race year SAT
1 Bob 21 70 180 Cauc Jr 1080
2 Fred 18 67 156 Af.Am Fr 1210
3 Barb 18 64 128 Af.Am Fr 840
4 Sue 24 66 118 Cauc Sr 1340
5 Jeff 20 72 202 Asian So 880
> midataframe$age
[1] 21 18 18 24 20
> midataframe[1,5]
[1] Cauc
Levels: Af.Am Asian Cauc
> midataframe[3,4]
[1] 128
> hist(midataframe$age)
>
```

Environment History

Import Dataset List

Global Environment

object	description
dataframe	3 obs. of 3 variables
m	num [1:2, 1:3] 1 2 3 4 5 6
m2	num [1:2, 1:3] 1 4 2 5 3 6
midataframe	5 obs. of 7 variables
sexoedad	10 obs. of 4 variables

Values

a	num [1:3] 11 12 13
---	--------------------

Files Plots Packages Help Viewer

Zoom Export Publish

Histogram of midataframe\$age

The histogram displays the distribution of ages from 18 to 24. The x-axis is labeled 'midataframe\$age' and ranges from 18 to 24. The y-axis is labeled 'Frequency' and ranges from 0.0 to 3.0. There are two bars: one for the range 18-20 with a frequency of 2.0, and another for the range 20-22 with a frequency of 1.0.

R. Cargar datos de archivos

Normalmente tenemos grandes cantidades de datos en archivos (texto, Excel, CSV, etc).

R puede leer ficheros con multitud de formatos: texto (extensión .txt), CSV (extensión .csv), Excel (con extensión .xls o.xlsx), SPSS (con extensión .sav), etc.

Para leer un fichero:

.txt => `read.table()`

.csv => `read.csv()`

.xls => `readWorksheet()`

Se obtiene un data frame con todas las observaciones de todas las variables incluidas en el fichero de texto.



R. Cargar datos de archivos

Leer un archivo de texto (primera línea como cabecera)

```
dataTXT <- read.table( "f.txt" , header=T )
dataTXT

dataTXT[ 1 ]

sin( dataTXT[ 1 ] )
cos( dataTXT[ 1 ] )
```



R. Cargar datos de archivos

Leer un archivo CSV (primera línea como cabecera)

```
dataCSV <- read.csv("./funcmat.csv")
dataCSV

dataCSV[1]

sin( dataCSV[1] )
cos( dataCSV[1] )
```

R. Cargar datos de archivos

Leer un archivo de Excel.

Primero, instalar el paquete necesario para poder leer archivos Excel.

```
install.packages('XLConnect',  
                 repos='http://cran.us.r-project.org')  
  
library(XLConnect)  
dataXLSX <- readWorksheet(  
    loadWorkbook("./funcmat.xlsx") , sheet=1 )
```

R. Cargar datos de archivos

Ejemplo:

Leer el archivo de ejemplo “sexoedad.txt”

Calcular la media de la columna edad.

Calcular la mediana de la columna altura.

Sexo	Edad	Peso	Altura
H	35	79	187
M	47	53	166
H	53	68	179
M	42	58	170
H	21	65	175
M	32	63	169
H	42	70	174
M	63	67	165
H	34	78	185
M	48	55	178



R. Cargar datos de archivos

Solución:

```
dataTXT <- read.table( "./sexoedad.txt" , header=T )  
  
# calcular la media de la columna "Edad"  
mean(dataTXT$Edad)  
  
# calcular la mediana de la columna altura  
median(dataTXT$Altura)  
  
# representar gráficamente la columna de "Edad"  
plot( dataTXT$Edad , type="l")
```



R. Cargar datos de archivos

RStudio

Project: (None)

funcmat sexoedad.txt ejemplos_presentacion.R

Source on Save Run Source

119 # ----- transp. 58
120 setwd("/Users/pedro/ejemplos/")
121 dataTXT <- read.table("./sexoedad.txt" , header=T)
122
123 # calcular la media de la columna edad
124 mean(dataTXT\$Edad)
125
126 # calcular la mediana de la columna altura
127 median(dataTXT\$Altura)
128
129 plot(dataTXT\$Edad , type="l")
130
131

(Top Level) R Script

Console Terminal ~/ejemplos/

```
>  
> setwd("/Users/pedro/ejemplos/")  
> dataTXT <- read.table( "./sexoedad.txt" , header=T )  
>  
> # calcular la media de la columna edad  
> mean(dataTXT$Edad)  
[1] 41.7  
>  
> # calcular la mediana de la columna altura  
> median(dataTXT$Altura)  
[1] 174.5  
>  
> plot( dataTXT$Edad , type="l")  
>
```

Environment History Connections

Import Dataset Global Environment

Data

dataTXT	10 obs. of 4 variables
funcmat	21 obs. of 5 variables

Values

real	4
x	num [1:63] 0 0.1 0.2 0.3 0.4 0...
y	num [1:63] 0 0.0998 0.1987 0.29...

Files Plots Packages Help Viewer

Zoom Export

Environment History Connections

Import Dataset Global Environment

Data

dataTXT	10 obs. of 4 variables
funcmat	21 obs. of 5 variables

Values

real	4
x	num [1:63] 0 0.1 0.2 0.3 0.4 0...
y	num [1:63] 0 0.0998 0.1987 0.29...

Files Plots Packages Help Viewer

Zoom Export

plot(dataTXT\$Edad , type="l")

Index	dataTXT\$Edad
1	35
2	48
3	53
4	45
5	20
6	35
7	45
8	60
9	35
10	45



R. Cargar datos de archivos

RStudio

ejemplos_presentacion.R*

```
107 # ----- transp. 54
108 setwd("./Users/pedro/ejemplos/")
109 dataTXT <- read.table( "./sexoedad.txt" , header=T )
110 
111 # calcular la media de la columna edad
112 mean(dataTXT$Edad)
113 
114 # calcular la mediana de la columna altura
115 median(dataTXT$Altura)
116 
117
```

116:1 (Top Level) R Script

Console ~/ejemplos/

```
> setwd("./Users/pedro/ejemplos/")
> dataTXT <- read.table( "./sexoedad.txt" , header=T )
> 
> # calcular la media de la columna edad
> mean(dataTXT$Edad)
[1] 41.7
> 
> # calcular la mediana de la columna altura
> median(dataTXT$Altura)
[1] 174.5
>
```

Environment History

Import Dataset

Global Environment

dataframe	3 obs. of 3 variables
dataTXT	10 obs. of 4 variables
m	num [1:2, 1:3] 1 2 3 4 5 6
m2	num [1:2, 1:3] 1 4 2 5 3 6
midataframe	5 obs. of 7 variables
sexoedad	10 obs. of 4 variables

Values

Files Plots Packages Help Viewer

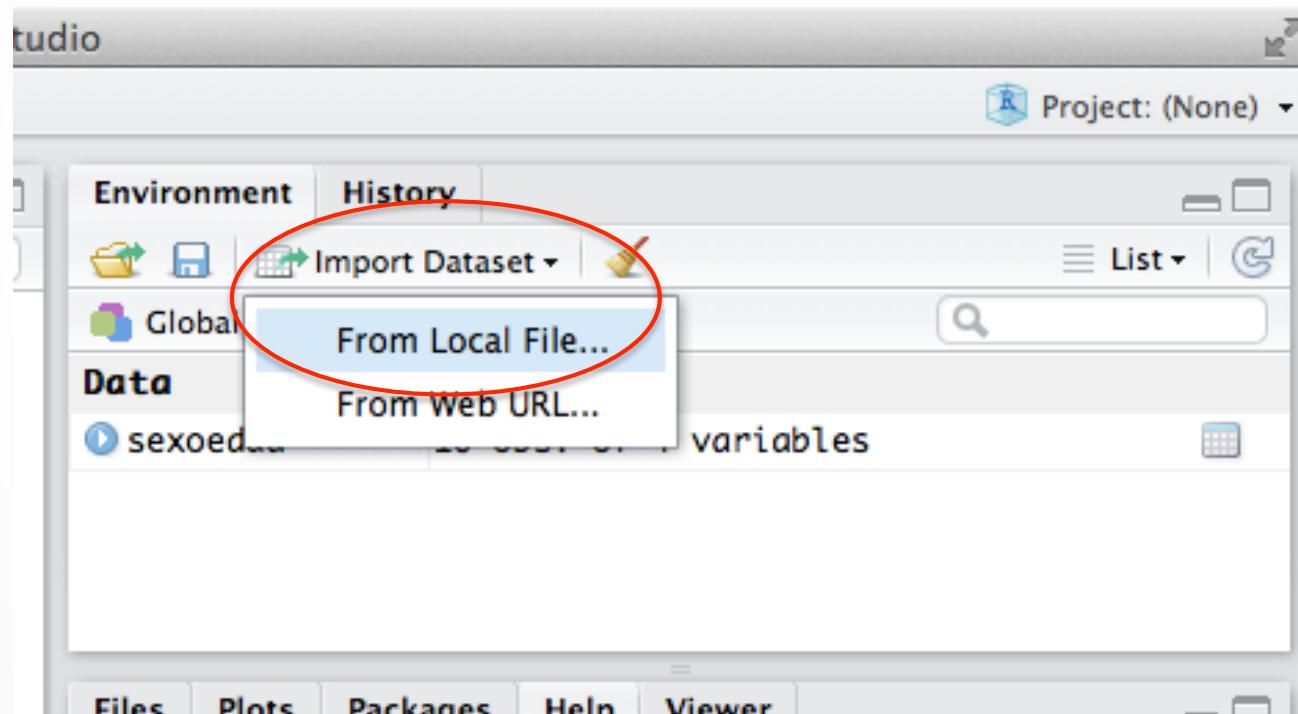
New Folder Delete Rename More

Home > ejemplos

..	acs_or.csv	721.5 KB	Feb 15, 2016, 11:35 PM
	ANOVA		
	ejemplos_presentacion.R	1.8 KB	Mar 5, 2016, 11:57 PM
	ejRnw		
	example01.txt	230 B	Feb 12, 2016, 10:21 AM
	funcmat.csv	716 B	Feb 12, 2016, 12:33 PM
	funcmat.txt	717 B	Feb 12, 2016, 12:35 PM

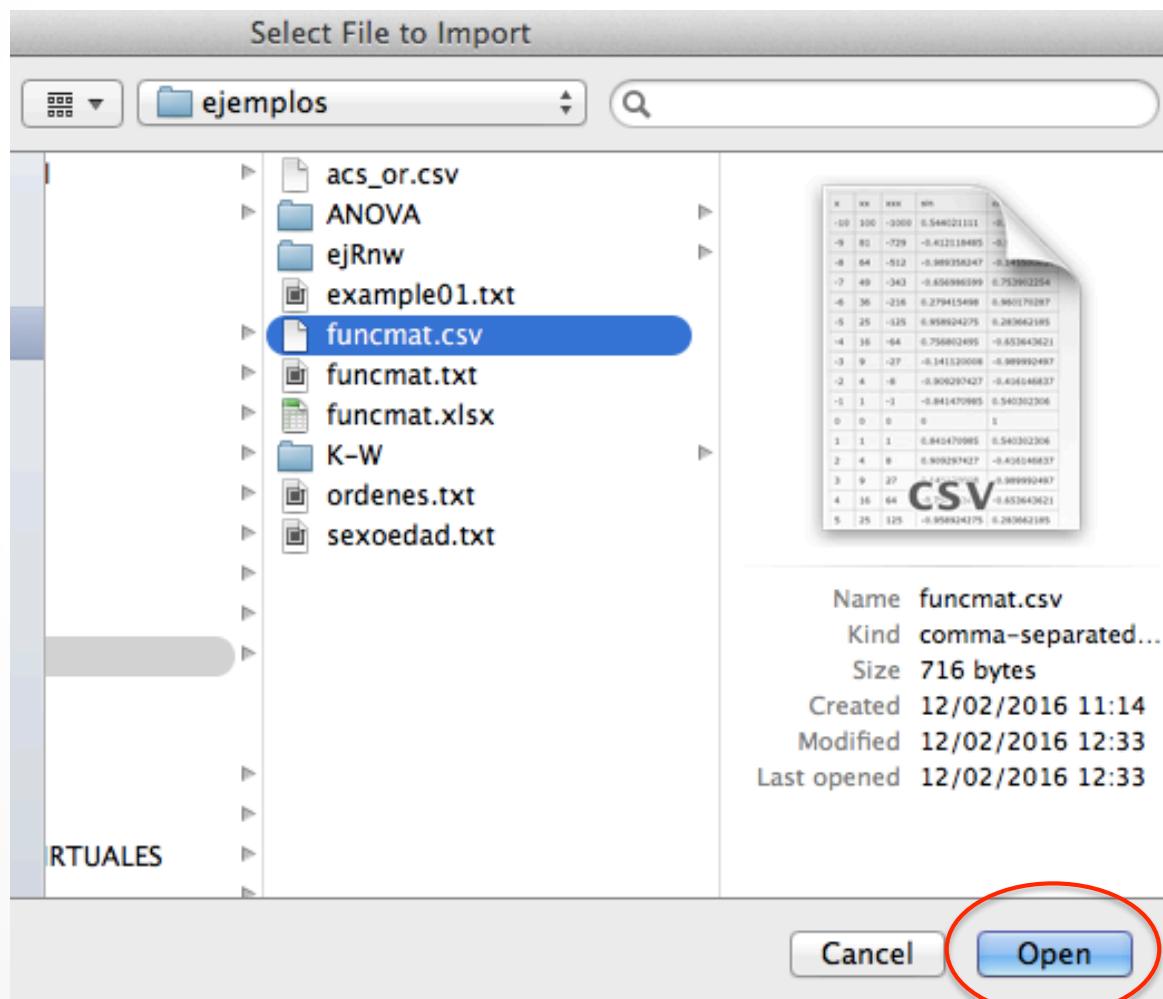
R. El espacio de trabajo

También podemos importar conjuntos de datos usando la interfaz gráfica:



R. El espacio de trabajo

Se nos pedirá que elijamos el archivo que tiene los datos:



R. El espacio de trabajo

Antes de importar los datos, podemos elegir cómo hacerlo:

Import Dataset

Name	funcmat
Encoding	Automatic
Heading	<input checked="" type="radio"/> Yes <input type="radio"/> No
Row names	Automatic
Separator	Comma
Decimal	Period
Quote	Double quote ("")
Comment	None
na.strings	NA

Strings as factors

x	xx	xxx	sin	cos
-10	100	-1000	0.54402111	-0.839071529
-9	81	-729	-0.412118485	-0.911130262
-8	64	-512	-0.989358247	-0.145500034
-7	49	-343	-0.656986599	0.753902254
-6	36	-216	0.279415498	0.960170287
-5	25	-125	0.958924275	0.283662185
-4	16	-64	0.756802495	-0.653643621
-3	9	-27	-0.14112008	-0.989992497
-2	4	-8	-0.909297427	-0.416146837
-1	1	-1	-0.841470985	0.540302306
0	0	0	0.0000000	1.0000000
1	1	1	0.8414710	0.5403023
2	4	8	0.909297427	-0.416146837

Import Cancel

R. El espacio de trabajo

The screenshot shows the RStudio interface with the following components:

- Top Bar:** Contains icons for file operations (New, Open, Save, Print), Go to file/function, Addins, and a Project dropdown set to (None).
- Environment Pane:** Shows the Global Environment with two datasets:
 - funcmat:** 21 obs. of 5 variables.
 - sexoedad:** 10 obs. of 4 variables.
- Data View:** A grid view showing columns x, xx, xxx, sin, and cos. The first 9 rows of the funcmat dataset are displayed, with the 10th row partially visible.
- Console:** Displays R command history and output. Key commands shown include:

```
>
>
> View(funcmat)
> funcmat$x
[1] -10 -9 -8 -7 -6 -5 -4 -3 -2 -1  0  1  2
[14]  3   4   5   6   7   8   9  10
> sin( funcmat$x )
[1]  0.5440211 -0.4121185 -0.9893582 -0.6569866
[5]  0.2794155  0.9589243  0.7568025 -0.1411200
[9] -0.9092974 -0.8414710  0.0000000  0.8414710
[13]  0.9092974  0.1411200 -0.7568025 -0.9589243
[17] -0.2794155  0.6569866  0.9893582  0.4121185
[21] -0.5440211
>
>
```
- Files Pane:** Shows the directory structure under ~/ejemplos/K-W/. The contents are:

Name	Size	Modified
..		
acs_or.csv	721.5 KB	Feb 15, 2016, 11:35 PM
ANOVA		
ejRnw		
example01.txt	230 B	Feb 12, 2016, 10:21 AM
funcmat.csv	716 B	Feb 12, 2016, 12:33 PM
funcmat.txt	717 B	Feb 12, 2016, 12:35 PM
funcmat.xlsx	44.6 KB	Feb 12, 2016, 11:14 AM
K-W		
ordenes.txt	4 KB	Mar 3, 2016, 11:29 PM
sexoedad.txt	142 B	Feb 14, 2016, 12:59 AM



Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?



Análisis de datos con R

R ofrece las funciones estadísticas más comunes.

A continuación veremos cómo aplicar diversas operaciones estadísticas a conjuntos de datos:

- | | |
|-----------------------------------|---------------------|
| • <code>mean(x)</code> | Media |
| • <code>median(x)</code> | Mediana |
| • <code>sd(x)</code> | Desviación estándar |
| • <code>var(x)</code> | Varianza |
| • <code>quantile(x, probs)</code> | Cuantiles |
| • <code>min(x)</code> | Mínimo |
| • <code>max(x)</code> | Máximo |

Análisis de datos con R

Como ejemplo, vamos a crear un vector de números. Calcularemos varias funciones estadísticas básicas, y representaremos el histograma:

```
x <- c(7, 4, 5, 9, 5, 7, 1, 5, 1)

max(x)
min(x)

mean(x)
sd(x)
median(x)

hist(x)
```





Análisis de datos con R

RStudio

Project: (None)

sexoedad * acs * Untitled1* x

Source on Save Import Dataset

Addins

Environment History

Global Environment

Values

x num [1:9] 7 4 5 9 5 7 1 5 1

Files Plots Packages Help Viewer

Zoom Export

Console ~/

R Script

```
3
4 x <- c(7, 4, 5, 9, 5, 7, 1, 5, 1)
5 sort(x)
6
7 max(x)
8 min(x)
9
10 mean(x)
11 sd(x)
12 median(x)
13
14 hist(x)
13:1 (Top Level) ▾
```

```
> x <- c(7, 4, 5, 9, 5, 7, 1, 5, 1)
> sort(x)
[1] 1 1 4 5 5 5 7 7 9
>
> max(x)
[1] 9
> min(x)
[1] 1
>
> mean(x)
[1] 4.888889
> sd(x)
[1] 2.666667
> median(x)
[1] 5
> hist(x)
> |
```

Histogram of x

A histogram titled "Histogram of x" showing the frequency distribution of the values in vector x. The x-axis is labeled "x" and ranges from 0 to 10 with major ticks at 0, 2, 4, 6, 8, and 10. The y-axis is labeled "Frequency" and ranges from 0.0 to 3.0 with major ticks at 0.0, 1.5, and 3.0. The histogram consists of five bars. The first bar (0-1) has a frequency of approximately 1.7. The second bar (1-2) has a frequency of approximately 0.3. The third bar (2-3) has a frequency of approximately 1.3. The fourth bar (3-4) has a frequency of approximately 3.0. The fifth bar (4-5) has a frequency of approximately 1.7. There are no bars for values 6, 7, 8, 9, and 10.

63

Distribuciones en R

R ofrece las funciones de probabilidad básicas:

```
help(Distributions)
```

Para cada tipo de distribución, un prefijo en el nombre indica la funcionalidad:

- "d" densidad
- "p" función de distribución
- "q" función cuantil
- "r" generación aleatoria de valores

Distribuciones en R

Ejemplo: crear un vector de cien números que siguen una distribución $N(10,2)$ y vamos a dibujar el histograma:

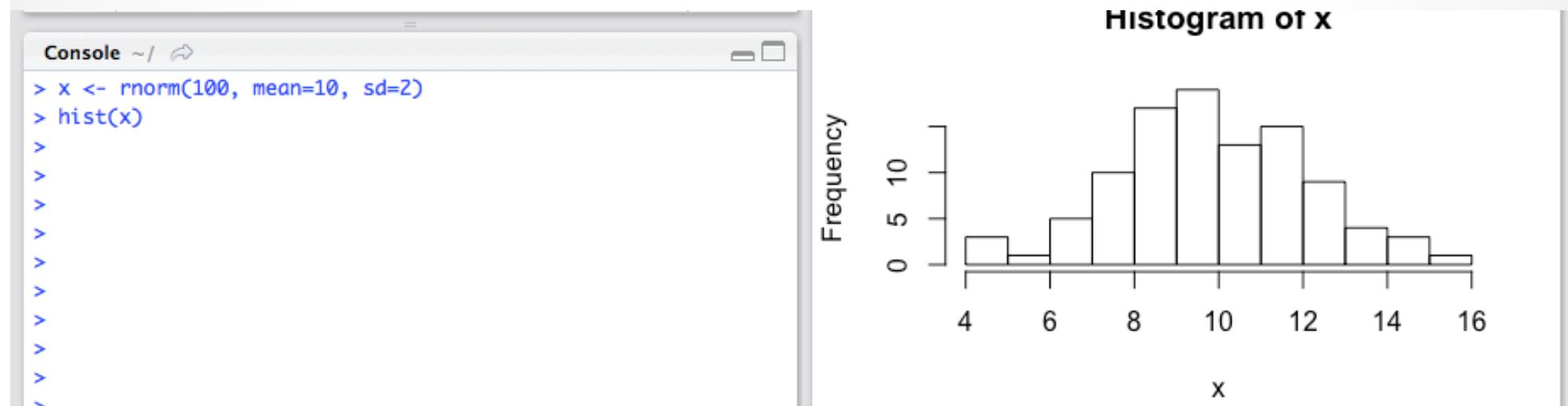
```
x <- rnorm(100, mean=10, sd=2)  
  
hist(x)
```



Distribuciones en R

Solución:

Como ejemplo, vamos a crear un vector de cien números que siguen una distribución $N(10,2)$ y vamos a dibujar el histograma:



Test estadísticos

Test estadísticos avanzados para comparar varios grupos de datos independientes entre sí.

Ver si diferentes niveles en una variable afectan significativamente a la medida:

- **ANOVA**

$$\sum_{i=1}^n (Y_i - \bar{Y})^2 = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 + PC$$

- **KRUSKAL-WALLIS**

$$H = \frac{\frac{12}{N(N+1)} \sum_{i=1}^k N_i \left(\bar{R}_i - \frac{(N+1)}{2} \right)^2}{1 - \frac{\sum_{j=1}^m (T_j^3 - T_j)}{(N^3 - N)}}$$

Test estadísticos

Paramétrico: ANOVA

ANOVA de un factor se usa cuando hay una variable-factor (nominal) y una medida (respuesta) que sigue una distribución normal.

El test nos dirá si diferentes niveles en el factor afectan significativamente a la respuesta (medida).

Test paramétrico ANOVA

Ejemplo:

```
dataTXT <- read.table( "./datos.txt" , header=T )  
  
anova( lm( response~method , dataTXT ) )  
  
TukeyHSD( aov( response~method , dataTXT ) )
```





Test paramétrico ANOVA

RStudio

Project: (None)

sexooedad * acs * Untitled1* *

Source on Save | Import Dataset | Global Environment | List | R: Anova Tables | Find in Topic | R Documentation

Environment History

Console ~/ejemplos/ANOVA/

```
> dataTXT <- read.table( "./datos.txt" , header=T )
> anova( lm( response~method , dataTXT ) )
Analysis of Variance Table

Response: response
          Df    Sum Sq   Mean Sq F value    Pr(>F)
method      3 2508925426 836308475  53.638 < 2.2e-16 ***
Residuals 108 1683917072 15591825
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> TukeyHSD( aov( response~method , dataTXT ) )
Tukey multiple comparisons of means
  95% family-wise confidence level

Fit: aov(formula = response ~ method, data = dataTXT)

$method
        diff      lwr      upr     p adj
dos-cuatro -8147.429 -10901.272 -5393.5854 0.0000000
tres-cuatro  4624.714   1870.871  7378.5575 0.0001588
uno-cuatro   1647.679   -1106.165  4401.5217 0.4050322
tres-dos     12772.143  10018.300  15525.9860 0.0000000
uno-dos      9795.107   7041.264  12548.9503 0.0000000
uno-tres     -2977.036  -5730.879  -223.1925 0.0287041
```

Data

acs 7811 obs. of 14 variables

dataTXT 112 obs. of 2 variables

method : Factor w/ 4 levels "cuatro", "dos", ... : 4 4 4 ...

response: int 79546 73379 66236 68628 74280 68095 687...

Files Plots Packages Help Viewer

anova {stats}

Anova Tables

Description

Compute analysis of variance (or deviance) tables for one or more fitted model objects.

Usage

anova(object, ...)

Arguments

object an object containing the results returned by a model fitting function (e.g., lm or glm).

... additional objects of the same type.

Test estadísticos

No paramétrico: Kruskal–Wallis

Kruskal–Wallis es “similar” a ANOVA de un factor, pero sin necesidad de que los datos sean normales.

- si p-value > 0.05 => NO hay diferencias significativas entre los niveles de la variable nominal.
- si p-value < 0.05 => SI hay diferencias significativas entre los niveles de la variable nominal.

Test no paramétrico Kruskal-Wallis

Ejemplo:

```
library(pgirmess)
Datos <- read.table( "./datos.txt" , header=T )

kruskal.test(response~method , data = Datos )

kruskalmc(Datos$response, Datos$method)
```



Test no paramétrico Kruskal-Wallis

RStudio

sexoeedad * acs * Untitled1* * Datos *

method response

Showing 1 to 0 of 120 entries

Console ~/ejemplos/K-W/

```
>
> ?kruskal.test
>
> setwd("./..")
> setwd("./K-W")
> library(pgirmess)
>
> Datos <- read.table( "./datos.txt" , header=T )
>
> kruskal.test(response~method , data = Datos )
```

Kruskal-Wallis rank sum test

data: response by method
 Kruskal-Wallis chi-squared = 11.6678, df = 2,
 p-value = 0.002927

```
> kruskalmc(Datos$response, Datos$method)
Multiple comparison test after Kruskal-Wallis
p.value: 0.05
Comparisons
      obs.dif critical.dif difference
diftam-mismotam  25.975    18.62079     TRUE
diftam-tamadap   8.150    18.62079    FALSE
mismotam-tamadap 17.825    18.62079    FALSE
>
> View(Datos)
>
```

Environment History

Import Dataset

Global Environment

Datos 120 obs. of 2 variables
 method : Factor w/ 3 levels "diftam","mismotam",...: 2...
 response: int 148032 505984 1515648 8892288 1663424 1...

sexoeedad 10 obs. of 4 variables

Values

Files Plots Packages Help Viewer

R: Kruskal-Wallis Rank Sum Test Find in Topic

kruskal.test {stats}

R Documentation

Kruskal-Wallis Rank Sum Test

Description

Performs a Kruskal-Wallis rank sum test.

Usage

```
kruskal.test(x, ...)
```

```
## Default S3 method:
kruskal.test(x, g, ...)
```

```
## S3 method for class 'formula'
kruskal.test(formula, data, subset, na.action, ...)
```

Arguments



Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

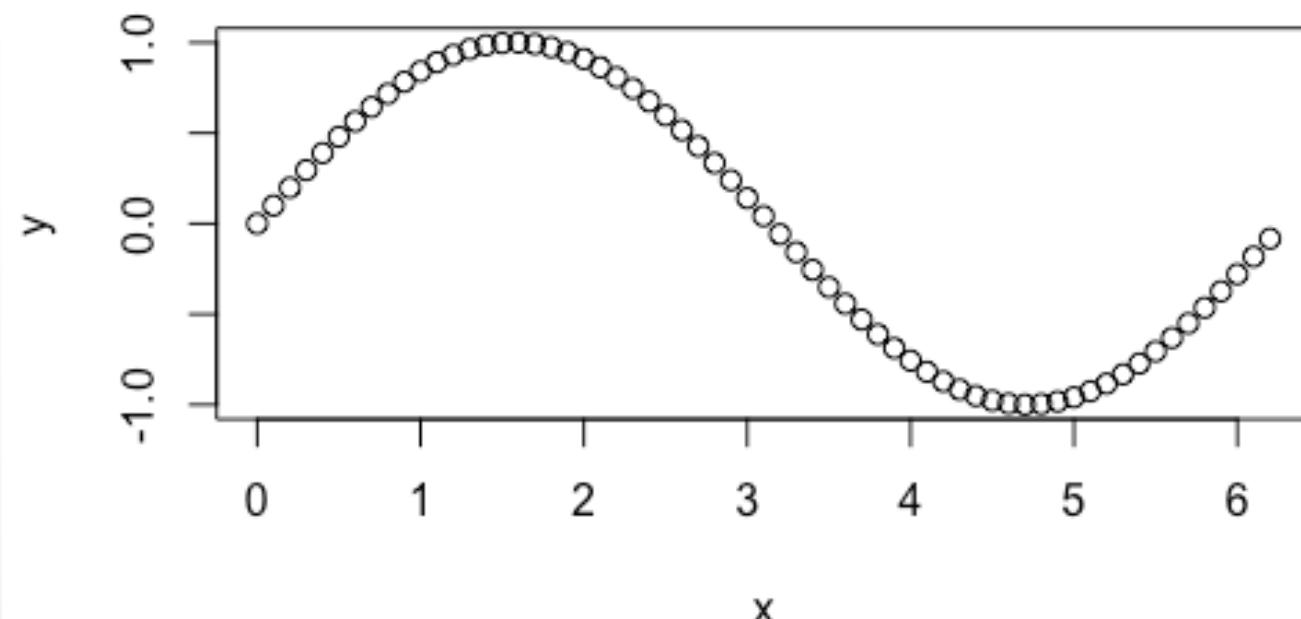
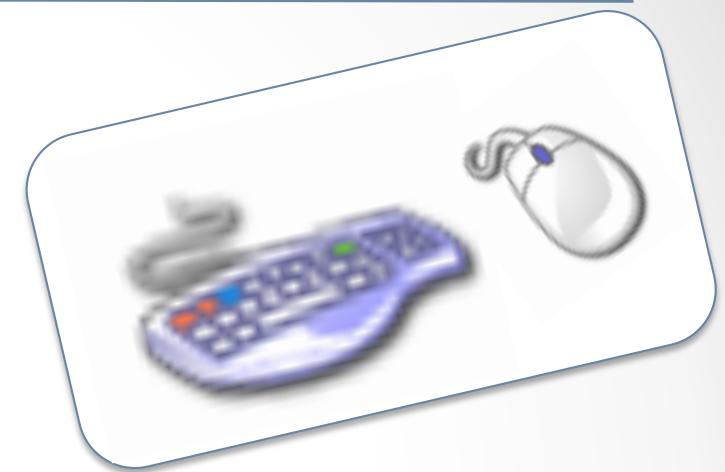
Programar en R

¿Qué más se puede hacer con R?



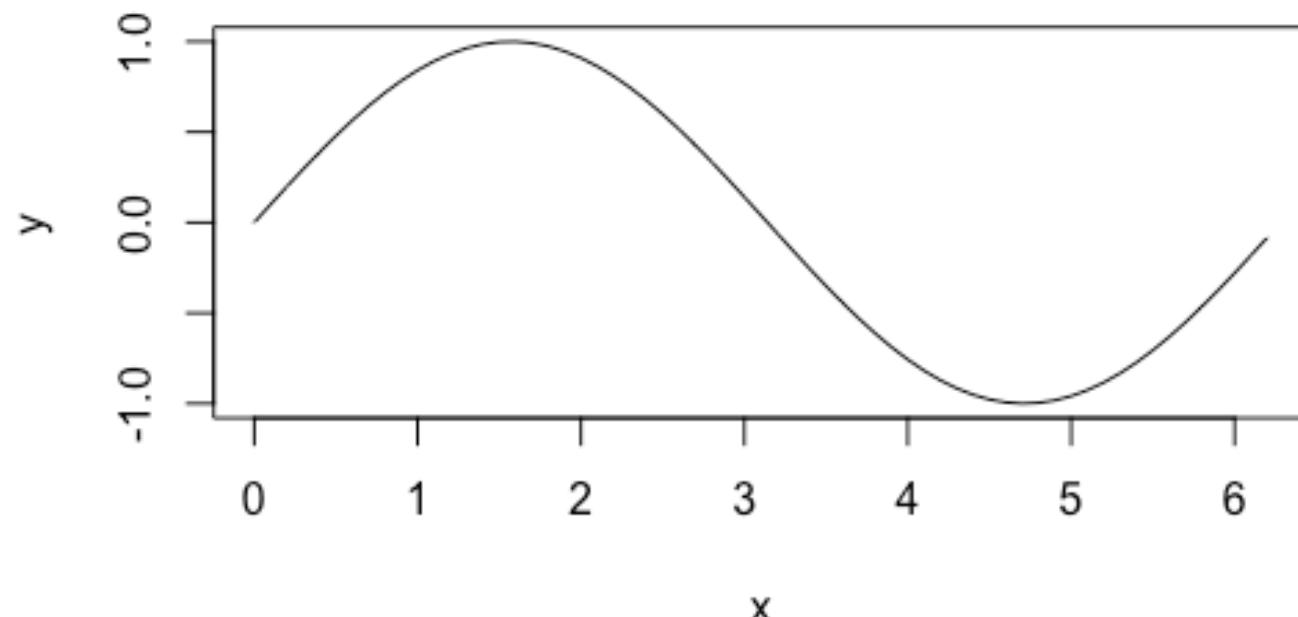
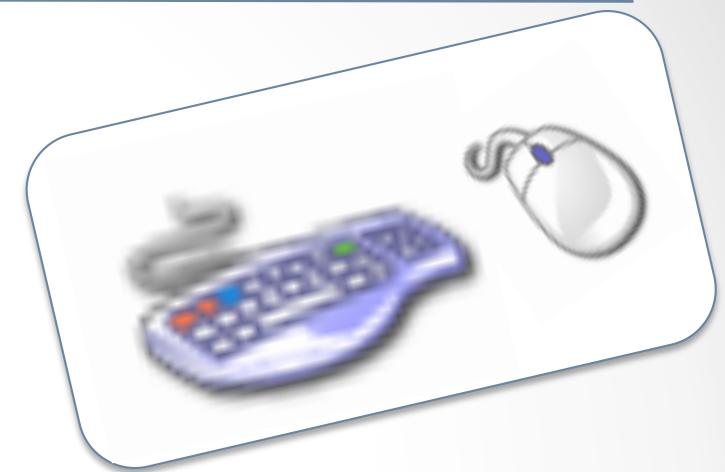
Gráficos con R

```
x <- seq(0,2*pi,0.1)
y <- sin(x)
plot(x,y,type="p")
```



Gráficos con R

```
x <- seq(0,2*pi,0.1)
y <- sin(x)
plot(x,y,type="l")
```





Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?



Programar en R

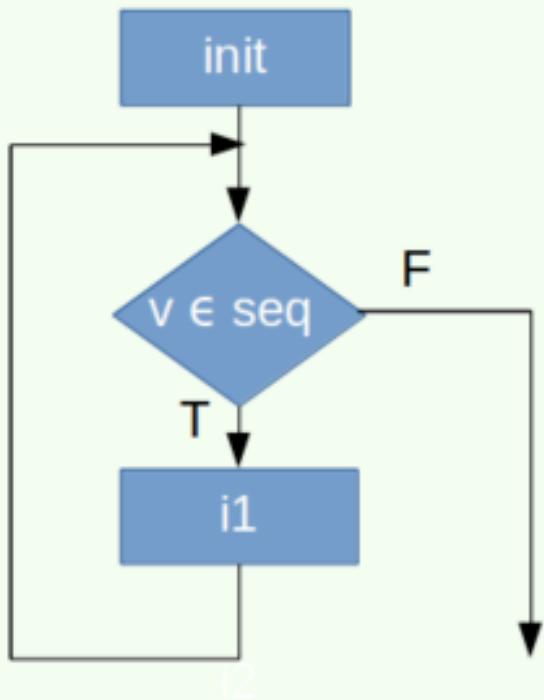
El lenguaje dispone de construcciones para controlar el flujo de ejecución:

- Bucles: for, while
- Condicionales: if-else
- Funciones

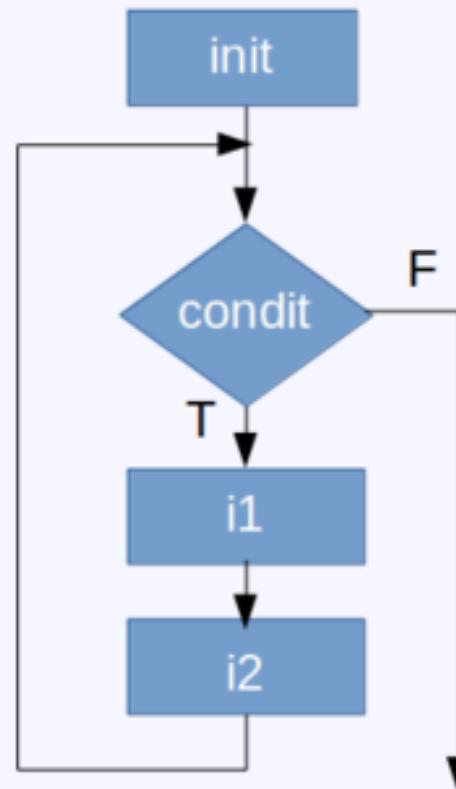
Programar en R. Bucles

Ejecutar repetidas veces un trozo de código:

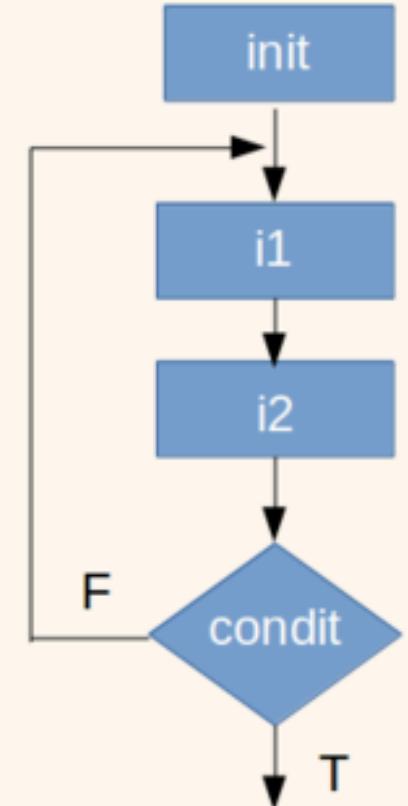
For loop



while loop



repeat loop



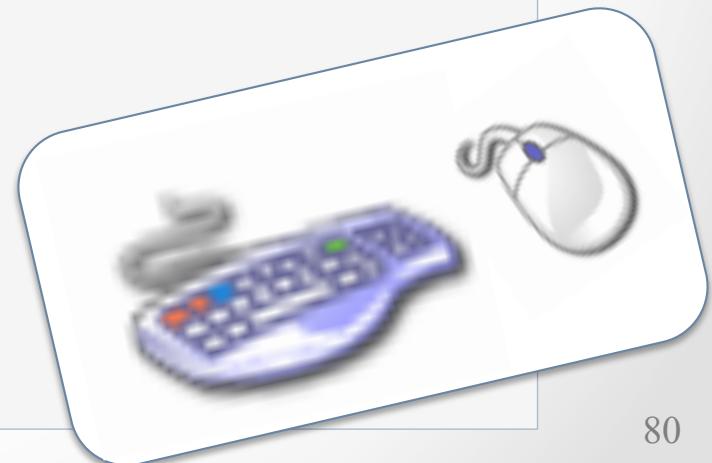
Programar en R. Bucles

Existen varios tipos de bucle:

```
for (i in 1:10)
  cat("El cuadrado de ", i, " es ", i^2, "\n")
```

```
mivector <- c(2,6,3,8,9,5)
for (v in mivector)
  cat("valor= ", v, "\n")
```

```
vector2 <- c()
mivector <- c(2,6,3,8,9,5)
for (i in 1:length(mivector))
  vector2 <- c(vector2, mivector[i])
vector2
```

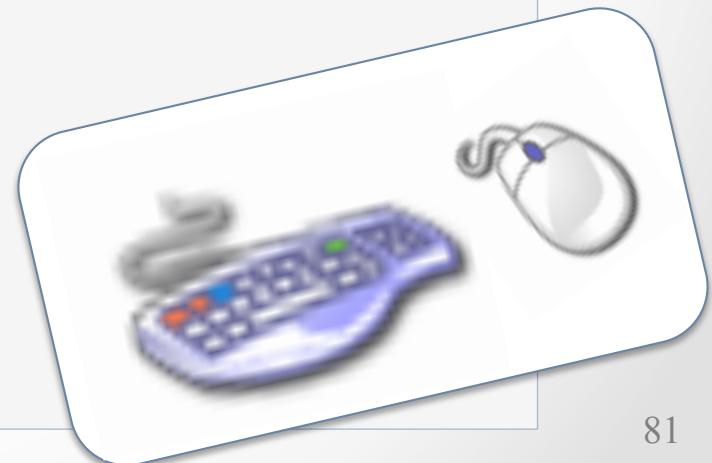


Programar en R. Condicional

Comprobar una condición para decidir si se ejecuta o no un trozo de código:

```
codigo <- 1
if(codigo==0)
    print("la variable vale cero")
```

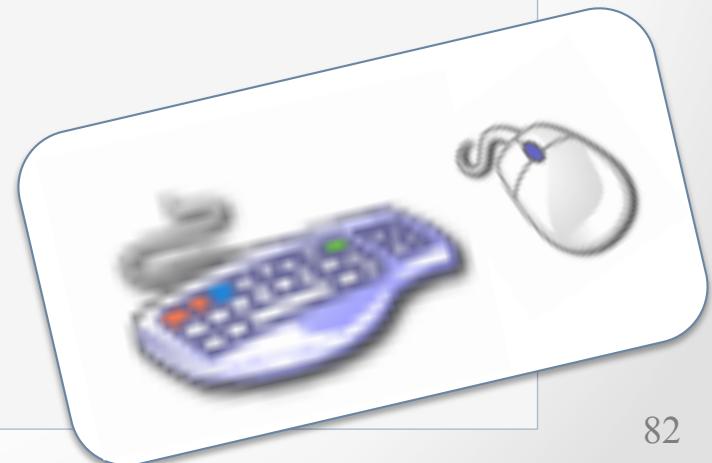
```
#leer de teclado y comprobar
codigo <- scan(n = 1)
if(codigo==7) {
    print("has acertado")
}else{
    print("has fallado")
}
```



Programar en R. Funciones

Podemos agrupar código que será llamado varias veces con diferentes valores:

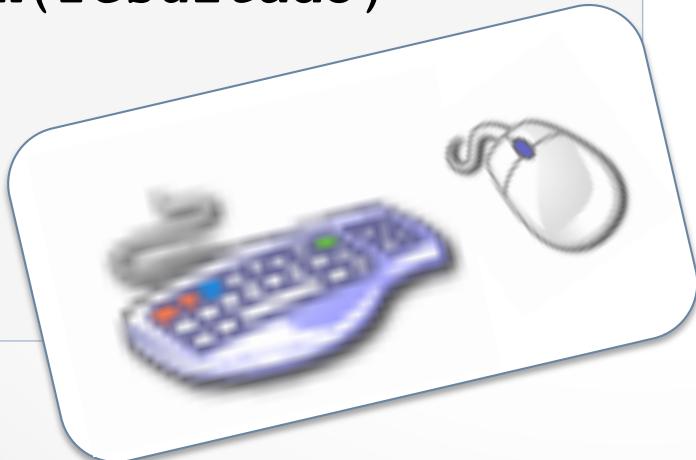
```
cuadrado <- function(x) {  
  square <- x * x  
  return(square)  
}  
  
ecuacion <- function(a,b,c) {  
  resultado <- a * b + c  
  return(resultado)  
}  
  
cuadrado(2)  
  
ecuacion(2,5,8)
```



Programar en R. Funciones

Podemos guardar las funciones en ficheros externos que podemos cargar:

```
cuadrado <- function(x) {  
  resultado <- x * x  
  return(resultado)  
}  
cubo <- function(x) {  
  resultado <- x * x * x  
  return(resultado)  
}
```



```
source("misfunciones.R")  
  
cubo(3)  
[1] 27  
  
cuadrado( cubo(2) )  
[1] 64
```

Calcular el tiempo de ejecución

La función **system.time()** calcula el tiempo que tarda un trozo de código en ejecutarse:

```
# crear un vector, y hacer un cálculo con
# cada elemento calculando el tiempo total

n <- 10000
sinI <- sample(1:n)

system.time(
  for(i in 1:n) sinI[i] <- sin(i)
)
```



Introducción a RStudio

Índice:

¿Qué es R?

¿Qué es RStudio?

Instalación de RStudio

Uso básico de RStudio

El lenguaje R

Análisis básico de datos con R

Representación gráfica de datos con R

Programar en R

¿Qué más se puede hacer con R?



¿Qué más se puede hacer con R?

R puede integrarse con distintas bases de datos.

Existen bibliotecas para usarlo desde Perl y Python.

Es posible desarrollar bibliotecas en C, C++ o Fortran que se cargan dinámicamente.

R puede usarse como herramienta de cálculo numérico, de forma similar a GNU-Octave o MATLAB.

RWeka7 permite interactuar con Weka para leer y escribir ficheros en el formato ARFF, y usar R con los algoritmos de minería de datos de Weka.

Se puede integrar código R en un texto en LaTeX con knitr.

Referencias

<http://www.clemson.edu/economics/faculty/wilson/R-tutorial/toc.html>

<http://www.cyclismo.org/tutorial/R/>

<https://cran.r-project.org/doc/contrib/R-intro-1.1.0-espanol.1.pdf>

"HSAUR: A Handbook of Statistical Analyses Using R (1st Edition)"

<https://cran.r-project.org/web/packages/HSAUR/index.html>

<http://www.harding.edu/fmccown/r/>

<http://www.datasciencecentral.com/profiles/blogs/10-great-books-about-r-1>

<http://www.datasciencecentral.com/page/search?q=r>

<https://www.rstudio.com/resources/cheatsheets/#515>

<https://cran.r-project.org/doc/contrib/grafi3.pdf>

Referencias

<http://adv-r.had.co.nz/Data-structures.html>

<http://hadley.nz/>

<http://r4ds.had.co.nz/>

Comparativa de lenguajes de programación (tipo R)

<https://www.datacamp.com/community/tutorials/statistical-language-wars-the-infograph>

Machine Learning in R for beginners

<https://www.datacamp.com/community/tutorials/machine-learning-in-r>

<https://www.datacamp.com/community/tutorials/tutorial-on-loops-in-r>

15 Questions All R Users Have About Plots

<https://www.datacamp.com/community/tutorials/15-questions-about-r-plots>

R Data Import Tutorial

<https://www.datacamp.com/community/tutorials/r-data-import-tutorial>

<https://www.datacamp.com/community/tutorials/importing-data-r-part-two>

Referencias

Blogs con recursos sobre R

<https://www.r-bloggers.com/>

<https://datascienceplus.com>

