

Introducción a la Programación en R con RStudio

estadisticalemo@delarivagroup.com

Temario

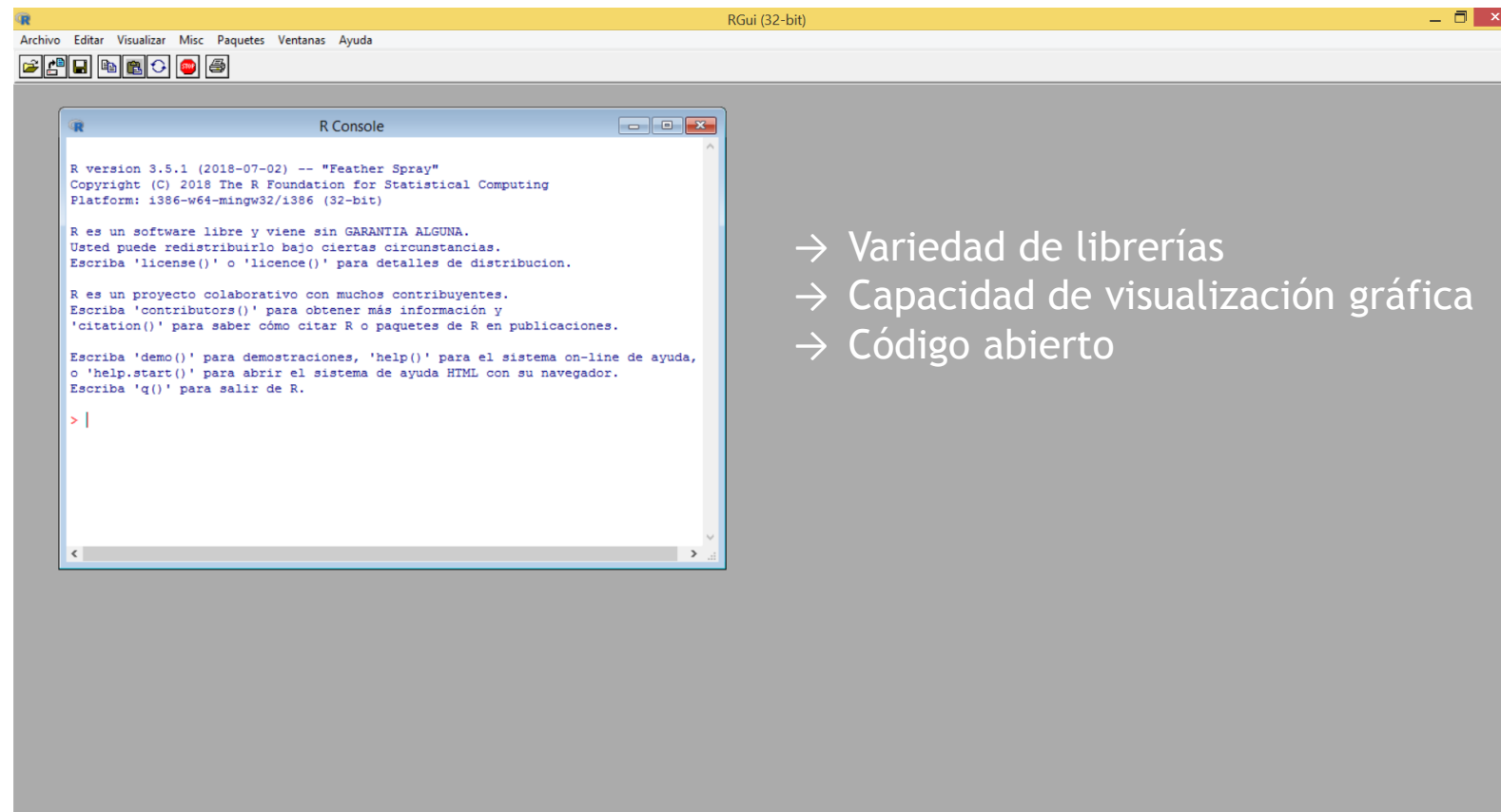
- 1) Entorno de desarrollo integrado (RStudio)**
- 2) Importación y manipulación de Data Frames**
- 3) Introducción a Tidyverse
- 4) Visualización de datos con ggplot2
- 5) Análisis exploratorio de datos
- 6) Creación de reportes

1. Entorno de desarrollo integrado



¿Qué es R?

R es un proyecto de software libre. Es tanto un lenguaje de programación como un entorno de trabajo, orientados al análisis estadístico principalmente. Sin embargo, los paquetes creados por los usuarios han ampliado su funcionalidad al procesamiento, análisis, modelación y reporte de datos entre otras cosas.



- Variedad de librerías
- Capacidad de visualización gráfica
- Código abierto



¿Qué es RStudio?

RStudio es un entorno de desarrollo integrado (amigable) para R. Puede ejecutarse sobre distintas plataformas (Windows, Mac, or Linux) o incluso desde la web usando RStudio Server.

The screenshot shows the RStudio IDE interface with several components labeled in red text:

- EDITOR**: pantalla donde se escriben las instrucciones (script)
- AMBIENTE**: pantalla donde se visualizan las variables declaradas
- CONSOLA**: pantalla donde se ejecutan comandos y se muestran los resultados
- GRAFICOS / PAQUETES / AYUDA**: This label points to the bottom-right pane, which currently displays the 'Packages' tab showing a list of installed and available R packages.

Name	Description	Version
<input type="checkbox"/> tibble	Simple Data Frames	1.4.2
<input checked="" type="checkbox"/> tidyr	Easily Tidy Data with 'spread()' and 'gather()' Functions	0.8.1
<input type="checkbox"/> tidyselect	Select from a Set of Strings	0.2.4
<input type="checkbox"/> tiff	Read and write TIFF images	0.1-5
<input type="checkbox"/> translations	The R Translations Package	3.5.1
<input type="checkbox"/> trimcluster	Cluster analysis with trimming	0.1-2
<input type="checkbox"/> utf8	Unicode Text Processing	1.1.4
<input type="checkbox"/> utils	The R Utils Package	3.5.1
<input type="checkbox"/> viridis	Default Color Maps from 'matplotlib'	0.5.1
<input type="checkbox"/> viridisLite	Default Color Maps from 'matplotlib' (Lite Version)	0.3.0
<input type="checkbox"/> whisker	{{mustache}} for R, logidess templating	0.3-2
<input type="checkbox"/> withr	Run Code 'With' Temporarily Modified Global State	2.1.2
<input type="checkbox"/> xtable	Export Tables to LaTeX or HTML	1.8-2
<input type="checkbox"/> yaml	Methods to Convert R Data to YAML and Back	2.1.19
<input type="checkbox"/> zip	Cross-Platform 'zip' Compression	1.0.0



Editor

The screenshot displays the RStudio environment. The **Tools** menu is open, with **Global Options...** selected. The **Run** button in the toolbar is also circled. The **Options** dialog box is open, showing the **Appearance** tab. The **Editor theme** list includes **Cobalt**, which is highlighted. The **Code** editor shows a function definition for `plot`.

Tools Menu:

- Install Packages...
- Check for Package Updates...
- Version Control
- Shell...
- Addins
- Keyboard Shortcuts Help Alt+Shift+K
- Modify Keyboard Shortcuts...
- Project Options...
- Global Options...**

Options Dialog - Appearance Tab:

- Zoom: 120%
- Editor font: Lucida Console
- Font size: 10
- Editor theme: Cobalt

Code Editor:

```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
    {
      !all(is.na(
        pmatch(names(list(...))
          "ylab")))
    }

    if (hasylab(...))
      plot.function(x, y, ...)
    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)",
          ...)
      )
  }
  else
    UseMethod("plot")
}
```



Consola & Ambiente



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1*

1 mean(x)
2 |

Run Source

Environment History

Global Environment

Values

x num [1:10] 1 2 3 4 5 6 7 8 9 10

2:1 (Top Level)

R Script

Files Plots Packages Help Viewer

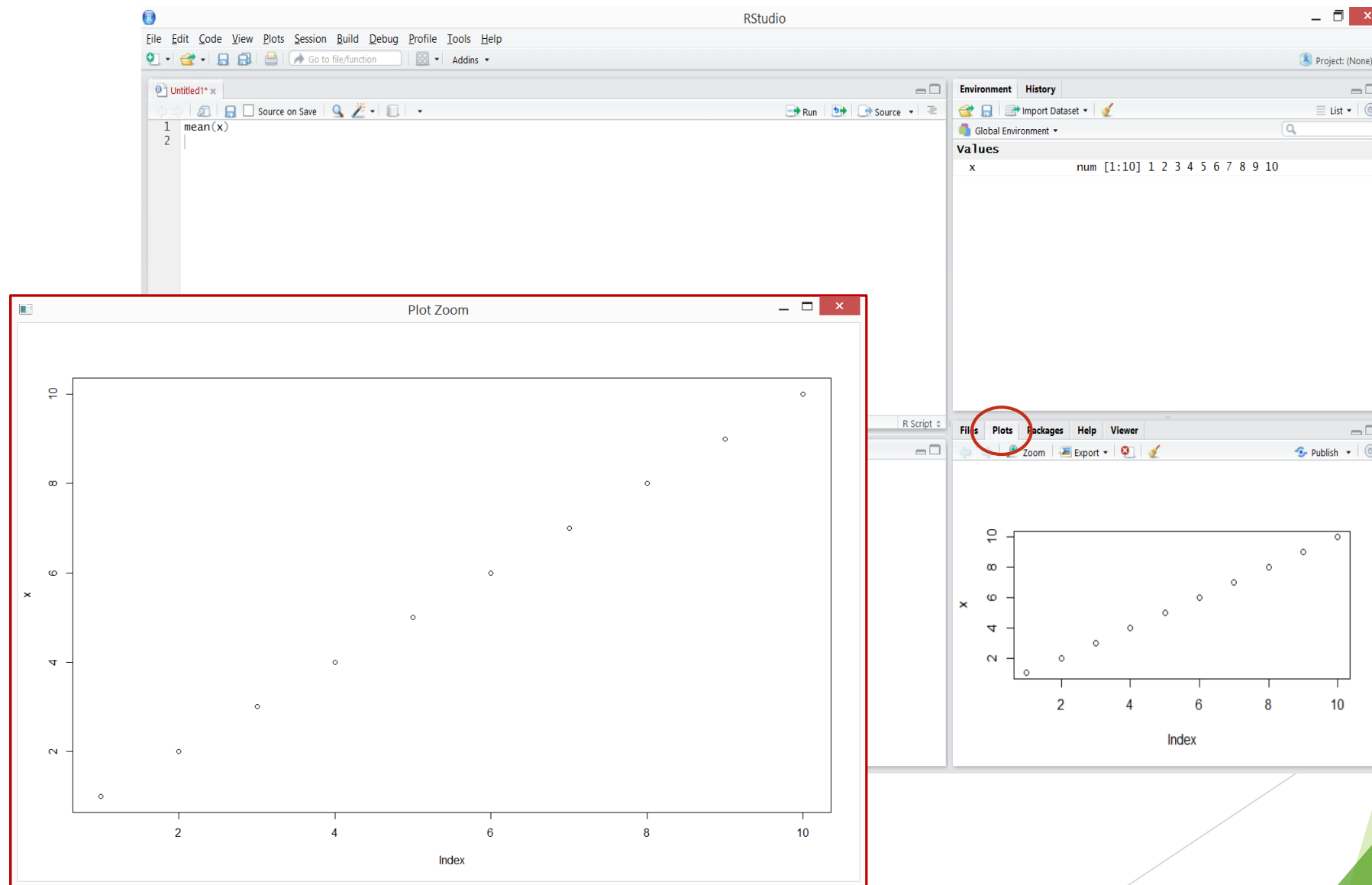
Zoom Export

Console //192.168.2.10/Lemonice/ESTADISTICA/Estudios/2020/NISSAN/12-0235-LI-MMT-19-00-TELEMATICS/

```
> x <- seq(1,10, by=1)
> x
[1] 1 2 3 4 5 6 7 8 9 10
> mean(x)
[1] 5.5
> |
```



Gráficas





Paquetes ¿Cómo cargar una librería?

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

introR.R x Untitled1 x Penalties_Fajitas.R x

Source on Save Run Source

1

Console

```
//192.168.2.18/Lemonice/ESTADISTICA/Estudios/2018/General Motors/04-0151-LI-DVO-18-00 WTP
122 3010 CAR B ROTACION2 WGRIDRun_r2_c5 500 2
123 3026 CAR B ROTACION1 WGRIDRun_r2_c5 150 2
124 3030 CAR B ROTACION1 WGRIDRun_r2_c5 200 2
125 3091 CAR B ROTACION1 WGRIDRun_r2_c5 120 2
[ reached getOption("max.print") -- omitted 1625 rows ]
> x
Error: object 'x' not found
> x<-c(1,2,"a","b")
> x
[1] "1" "2" "a" "b"
> x<-c(1,2,"a","b",10:16)
> x
[1] "1" "2" "a" "b" "10" "11" "12" "13" "14" "15" "16"
> x[4]
[1] "b"
> x[c(2,5)]
[1] "2" "10"
> sample(x,size=3,replace=TRUE)
[1] "14" "2" "14"
> length(x)
[1] 11
>
```

Install Packages

Install from: [Configuring Repositories](#)

Repository (CRAN)

Packages (separate multiple with space or comma):

Install to library: C:/Users/eramirez/Documents/R/R-3.5.1/library [Default]

☒ Install dependencies

Install Cancel

Environment History

Global Environment

Data

base	493 obs. of 3424 variables
base_w5	50 obs. of 3 variables
datos	70 obs. of 228 variables

Values

i	228L
plat	"CostaRica_CarB"
x	chr [1:11] "1" "2" "a" "b" "10" "11" "12" "13" ...

Files Plots Packages Help Viewer

Install Update

	Name	Description	Version
<input type="checkbox"/>	table	Simple Data Frames	1.4.2
<input checked="" type="checkbox"/>	tidy	Easily Tidy Data with 'spread()' and 'gather()' Functions	0.8.1
<input type="checkbox"/>	tidyselect	Select from a Set of Strings	0.2.4
<input type="checkbox"/>	tiff	Read and write TIFF images	0.1-5
<input type="checkbox"/>	tools	Tools for Package Development	3.5.1
<input type="checkbox"/>	translations	The R Translations Package	3.5.1
<input type="checkbox"/>	trimcluster	Cluster analysis with trimming	0.1-2
<input type="checkbox"/>	utf8	Unicode Text Processing	1.1.4
<input type="checkbox"/>	utils	The R Utils Package	3.5.1
<input type="checkbox"/>	viridis	Default Color Maps from 'matplotlib'	0.5.1
<input type="checkbox"/>	viridisLite	Default Color Maps from 'matplotlib' (Lite Version)	0.3.0
<input type="checkbox"/>	whisker	((mustache)) for R, logidess templating	0.3-2
<input type="checkbox"/>	withr	Run Code 'With' Temporarily Modified Global State	2.1.2
<input type="checkbox"/>	xtable	Export Tables to LaTeX or HTML	1.8-2
<input type="checkbox"/>	yaml	Methods to Convert R Data to YAML and Back	2.1.19
<input type="checkbox"/>	zip	Cross-Platform 'zip' Compression	1.0.0



Ayuda

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains the code `mean(x)` on line 1 and 2.
- Environment:** Shows the Global Environment with a variable `x` of type `num` [1:10] containing values 1 through 10.
- Console:** Shows the execution of the following commands:

```
> x <- seq(1,10, by=1)
> x
[1] 1 2 3 4 5 6 7 8 9 10
> mean(x)
[1] 5.5
> plot(x)
```
- Help Panel:** Displays the documentation for `barplot`, which is highlighted with a red box. The documentation includes the title `barplot (graphics)`, a description, usage information, and the default S3 method signature.

The `barplot` documentation text is as follows:

```
barplot (graphics)
R Documentation

Bar Plots

Description
Creates a bar plot with vertical or horizontal bars.

Usage
barplot(height, ...)

## Default S3 method:
barplot(height, width = 1, space = NULL,
names.arg = NULL, legend.text = NULL, beside = FALSE,
horiz = FALSE, density = NULL, angle = 45,
col = NULL, border = par("fg").
```



Ayuda

The screenshot displays the RStudio interface with the following components:

- Script Editor:** Contains a script with two lines: `1 mean(x)` and `2`.
- Environment Pane:** Shows the 'Global Environment' with a variable `x` of type `num` (numeric) with values `[1:10]`.
- Console:** Shows the command `> help("read.csv")` entered, which is circled in red.
- Help Pane:** Displays the documentation for `read.table` (utils), which is enclosed in a red rectangle. The documentation includes the title `read.table (utils)`, the section `Data Input`, a `Description` stating it reads a file in table format and creates a data frame, and a `Usage` section with the function signature: `read.table(file, header = FALSE, sep = "", quote = "\"", dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"), row.names, col.names, as.is = !stringsAsFactors, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE,`



Ayuda

The screenshot shows the RStudio interface. The main editor window contains the following R code:

```
1 mean(x)
2
```

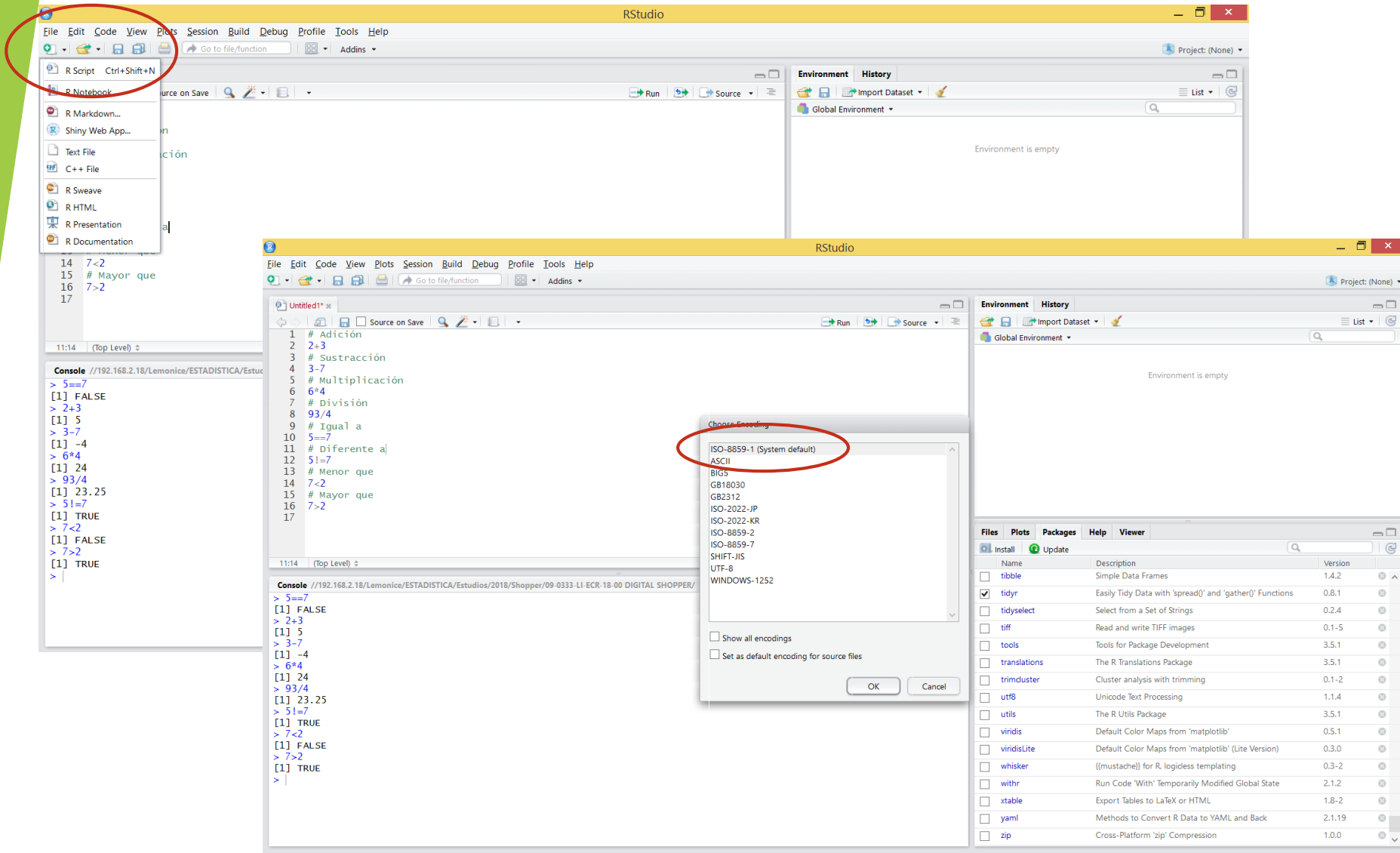
The console window at the bottom left shows the command `> ?setwd` entered, which has been circled in red. The console output shows the current working directory: `//192.168.2.10/Lenonice/ESTADISTICA/Estudios/2020/NISSAN/12-0235-LI-MMT-19-00-TELEMATICS/`.

The Environment pane on the right shows the Global Environment with a variable `x` of type `num` containing values `[1:10]`.

The Help pane on the bottom right, outlined in red, displays the documentation for the `setwd` function. The documentation includes the following sections:

- getwd (base)** R Documentation
- Get or Set Working Directory**
- Description**
`getwd` returns an absolute filepath representing the current working directory of the R process; `setwd (dir)` is used to set the working directory to `dir`.
- Usage**
`getwd ()`
`setwd (dir)`
- Arguments**
`dir` A character string; [file expansion](#) will be done.

¿Cómo guardar un proyecto (programa)?



2. Importación y manipulación de Data Frames

Asignación

En R, todos los datos y estructuras de datos son objetos a los cuales se les asigna un nombre (preferiblemente claro y descriptivo) para identificarlos.

```
<-  
# La declaración "apunta" hacia el objeto que recibe el valor de la expresión  
->
```

```
Ej.  n <- 4  
      7 -> n      # R es sensible al uso de mayúsculas y minúsculas
```

NOTA: si un objeto ya existe, su valor anterior es borrado después de la asignación. Si una expresión no es asignado a un objeto, el resultado se mostrará en pantalla pero no será guardado.

```
Ej.  n1 <- 5 + 3      # se asigna el resultado de la operación a la variable n1  
      (8*2)-1         # solo se muestra el resultado en la consola
```

Para correr el código se puede utilizar el botón "RUN" del Editor o el atajo "Ctrl + r".
Es posible ejecutar todo el código o sólo partes de él.

```
ls ()  # enlista los objetos en memoria
```

Comentarios

Esto es un comentario, no se ejecuta.

Sirve para describir el programa, lo cual es una buena práctica.

Cada línea debe empezar con el símbolo gato.

Eliminación

`rm()`

Ej. `rm(n1)`

Limpiar la consola

`Ctrl + l`

Tipo de Datos

Algunos de los tipos de dato más comunes son:

Cadena de texto	character	"hola"
Numérico	numeric	5.9
Entero	integer	2
Factor	factor	1 = hombre / 2 = mujer (variable categórica)
Lógico	logical	TRUE / FALSE
Perdido (faltante)	not available	NA
Vacío (no existente)	null	NULL
No Numérico	not a number	NaN

En R los datos pueden ser forzados a ser de otro tipo para ejecutar determinadas acciones. Las coerciones pueden ejecutarse en el siguiente orden (del más restrictivo al más flexible):

lógico -> entero -> numérico -> cadena de texto

Los factores pueden ser coercionados a tipo numérico y cadena de texto, sin embargo, es importante considerar que al forzar un factor a tipo numérico, se pierden sus niveles.

- `as.logical()`
- `as.integer()`
- `as.numeric()`
- `as.character()`
- `as.factor()`

*# si la coerción no puede realizarse, su ejecución
devolverá NA como resultado*

Operadores

Los operadores son símbolos que permiten realizar ciertas acciones. Los principales son:

ARITMETICOS

+	adición
-	sustracción
*	multiplicación
/	división
^	potencia

RELACIONALES

<	menor que
>	mayor que
<=	menor o igual que
>=	mayor o igual que
==	igual a
!=	diferente de

LOGICOS

X & Y	Y lógico
X Y	O lógico
! X	NO lógico
X && Y	idéntico
X Y	idéntico

Vectores

Un vector es una colección de uno o más datos del mismo tipo y es la estructura de datos más sencilla en R.

Propiedades de un vector:

- ✓ **Tipo.** Un vector tiene el mismo tipo que los datos que contiene. Si los datos del vector son de tipo numérico, el vector será también de tipo numérico.
- ✓ **Largo.** Es el número de elementos que contiene un vector.

<code>c(a,b,c)</code>	crea vector concatenando las letras a,b,c
<code>rep (5,n)</code>	repite el valor 5 n-veces
<code>seq(1:n)</code>	secuencia: 1, 2, 3, . . . , n
<code>seq(a,b,by=c)</code>	(a, a + c, a + 2c, . . .) hasta b
<code>x[4]</code>	selecciona la cuarta coordenada del vector x
<code>x[c(2,5)]</code>	selecciona la segunda y quinta coordenada del vector x[c(2,5)]
<code>sample(x,size=k,replace=TRUE)</code>	saca muestra de x, tamaño k, CON repetición
<code>sample(x,size=k,replace=FALSE)</code>	saca muestra de x, tamaño k, SIN repetición
<code>length(x)</code>	calcula la longitud del vector x

Data Frame

Un Data Frame o Base de Datos es una tabla compuesta de uno o mas vectores de la misma longitud pero que pueden ser de diferentes tipos.

¿Cómo establecer una ruta de trabajo?

```
setwd("C:/Users/eramirez/Documents/")  
getwd()
```

¿Cómo cargar una base a R?

Los paquetes o librerías son una colección de funciones con una estructura bien definida diseñadas para resolver una tareas específicas.

Aunque se cuente con un paquete instalado previamente, las sesiones de R se inician “limpias” por lo que cada sesión se deben importar las librerías requeridas.

```
library("foreign")
```

NOTA: con la función `installed.packages()` se pueden ver los paquetes instalados.

si se ha establecido previamente la ruta de trabajo

```
base1 <- read.csv("cars.csv", header = TRUE)
```

si NO se ha establecido previamente la ruta de trabajo

```
base2 <- read.csv('C:/Users/eramirez/Documents/cars.csv', header = TRUE)
```

```
base3 <- read.csv(file.choose(), header=TRUE)
```

Revisión de una base de datos

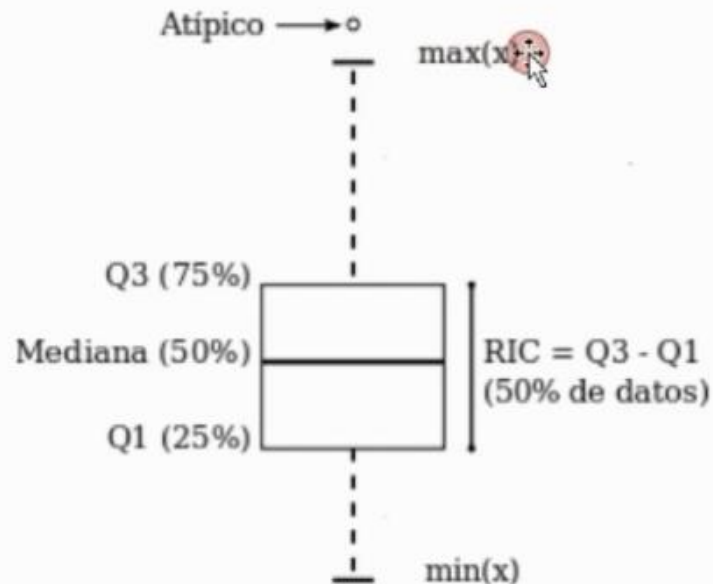
- `dim(base)` Muestra el tamaño o dimensión de una base
- `names(base)` Muestra los nombres de los campos (variables) de una base
- `str(base)` Muestra la estructura de datos en una base
- `head(base)` Muestra el encabezado de una base
- `tail(base)` Muestra los últimos registros de una base
- `summary(base)` Muestra un resumen estadístico de los campos de una base
- `table(base$campo)` Muestra una tabla de frecuencias del campo especificado
- `prop.table(base$campo)` Muestra una tabla de frecuencias del campo especificado como proporciones

Funciones matemáticas		Funciones estadísticas	
<code>sqrt(x)</code>	Raíz de x	<code>mean(x)</code>	Media
<code>exp(x)</code>	Exponencial de x	<code>sd(x)</code>	Desvio
<code>log(x)</code>	Logaritmo natural de x	<code>var(x)</code>	Varianza
<code>log10(x)</code>	Logaritmo base 10	<code>median(x)</code>	Mediana
<code>length(x)</code>	Número de elementos	<code>quantile(x,p)</code>	Quantiles
<code>sum(x)</code>	Suma los elementos de x	<code>max(x)</code>	El máximo
<code>prod(x)</code>	Producto de los elementos	<code>min(x)</code>	El mínimo
<code>sin(x)</code>	Seno	<code>summary(x)</code>	Resumen
<code>cos(x)</code>	Coseno	<code>sort(x)</code>	Ordena (creciente)
<code>tan(x)</code>	Tangente		
<code>round(x,n)</code>	redondea a n dígitos		
<code>cumsum(x)</code>	calcula las sumas acumuladas		
<code>choose(n, k)</code>	calcula en combinatorio		

Gráficos (básicos)

<code>plot(x,y)</code>	grafica los pares x vs. y
<code>hist(x)</code>	realiza un histograma con los valores del vector x
<code>boxplot(x)</code>	realiza un boxplot con los valores del vector x
<code>barplot(x)</code>	realiza gráfico de barras de x
<code>pie(table(x))</code>	realiza gráfico de torta de la tabla de x
<code>par(mfrow = c(1, 2))</code>	particiona el entorno gráfico

BOXPLOT



Q1 = primer cuartil

Q2 = segundo cuartil = mediana

Q3 = tercer cuartil

RIC = rango intercuartílico

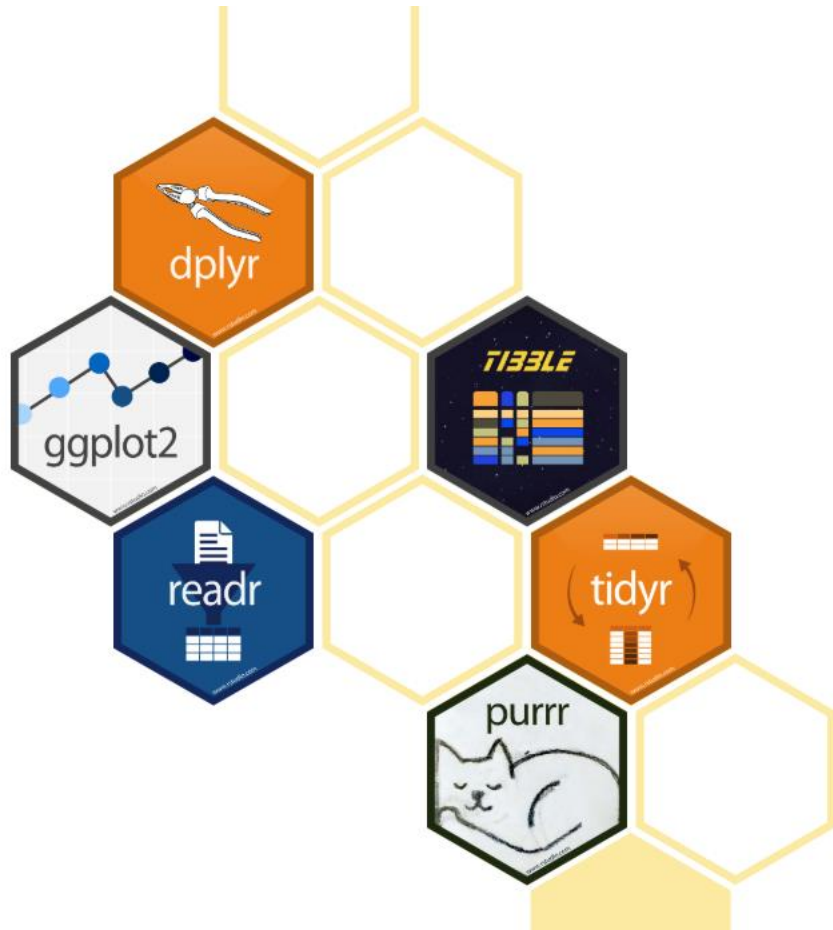
Los cuartiles son los tres valores de la variable que dividen los datos ordenados en cuatro partes iguales. Determinan los valores correspondientes al 25%, 50% (mediana), 75% de los datos.

Temario

- 1) Entorno de desarrollo integrado (RStudio)
- 2) Importación y manipulación de Data Frames
- 3) Introducción a Tidyverse
- 4) Visualización de datos con ggplot2
- 5) Análisis exploratorio de datos
- 6) Creación de reportes

3. Introducción a Tidyverse

Tidyverse



Es un conjunto de librerías de R diseñadas para la “*Ciencia de datos*”. Todas las librerías de este conjunto comparten la misma filosofía de trabajo, la misma gramática y las mismas estructuras de datos.

Instalación del paquete:

```
install.packages("tidyverse")
```

Librerías más comunes de Tidyverse:

ggplot2	visualización de datos
dplyr	manipulación de datos
tidyr	herramientas para ordenar datos
readr	para la importación de datos
tibble	manejo de data frames
stringr	para trabajar fácilmente con cadenas
rmarkdown	para programación reproducible