

**Lenguaje R y tratamiento de datos**  
**© EDICIONES ROBLE, S.L.**

# Indice

<b>Lenguaje R y tratamiento de datos</b>	<b>3</b>
I. Introducción	3
II. Objetivos específicos	3
III. Análisis estadístico	3
IV. Lenguaje R y RStudio	4
V. Comandos esenciales en R y tipos de dato	6
VI. Operaciones útiles sobre tablas, carga y descarga de datos	9
VII. Funciones y bucles	10
VIII. Resumen final	11
<b>Ejercicios</b>	<b>12</b>
Caso práctico	12
Solución	12
<b>Recursos</b>	<b>13</b>
Enlaces de Interés	13
Glosario.	13

# Lenguaje R y tratamiento de datos

## I. Introducción

El tratamiento de datos y su análisis es el objetivo último de las tecnologías Big Data. Todo el potencial se basa en que, al generar y almacenar eficientemente cantidades ingentes de información, los algoritmos matemáticos y estadísticos podrán minar y extraer patrones explotables científica o comercialmente.

De entre los lenguajes de datos, R es nativo para aplicar modelos estadísticos de alto nivel. Además, es sencillo de usar para programadores no expertos, por lo que es una buena elección para mostrar las aplicaciones de los conceptos estadísticos fundamentales.

En el resumen de esta misma unidad podrás encontrar un videotutorial con ejemplos.

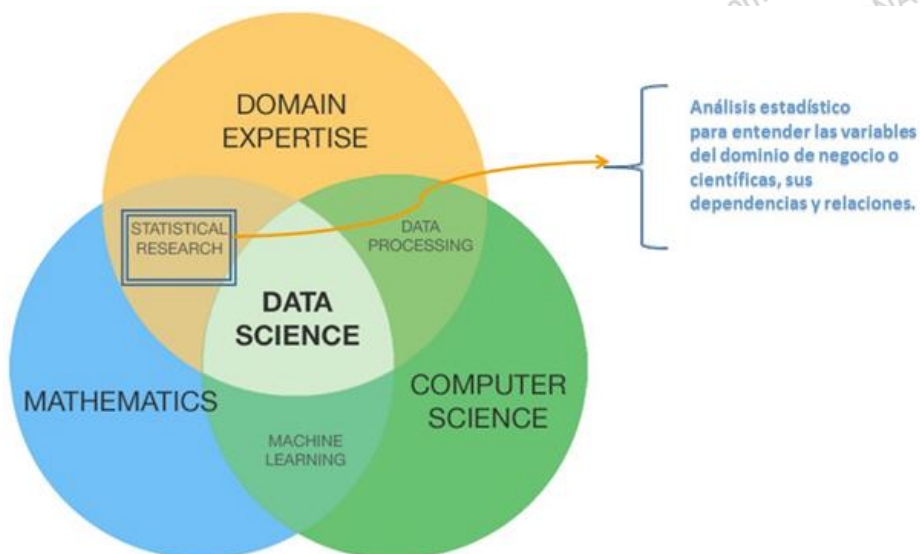
## II. Objetivos específicos



- Entender en qué consiste el lenguaje R y su IDE.
- Iniciarse con los notebooks.
- Trabajar con tablas de datos y conocer los comandos básicos de tratamiento de las mismas.
- Definir funciones en R, bifurcaciones y bucles.

## III. Análisis estadístico

El análisis estadístico es uno de los pilares en los que se sustenta la ciencia de datos. Aporta una explicación científica de la incertidumbre en los datos y ayuda a entender mejor los mismos dotándolos de estructura.



**Imagen 1.1.** Disciplinas. *Fuente:* elaboración propia.

## IV. Lenguaje R y RStudio

El análisis estadístico es una de las herramientas centrales en la ciencia de datos. Los modelos y algoritmos que se usan tienen su base en la estadística matemática, de modo que conocer los fundamentos en los que se basan es imprescindible para desarrollarlos y controlarlos.

El trabajo de la ciencia de datos se desarrolla casi íntegramente en el seno de la programación; ya que, para tratar con los datos, analizarlos y explotarlos, usando algoritmos a una escala media-alta, es inviable el análisis por inspección.

### Lenguaje R

Dentro de los lenguajes de programación dedicados a la estadística, uno de los más representativos y eficientes es R.



Imagen 1.2. R.

Fuente: [www.wikipedia.com](http://www.wikipedia.com)

R es un lenguaje diseñado para el tratamiento de datos y la aplicación de procesos y modelado estadístico. Este tiene muchas facilidades incorporadas en su sintaxis, multitud de librerías específicas para distintas tareas estadísticas, que a su vez incorporan una colección de datos copiosa. Además, se enmarca en lo que se conoce como **código abierto**: esto significa que todas las librerías, herramientas y código base se pueden usar gratuitamente.



La instalación del lenguaje R se realiza siguiendo las instrucciones que incluye su repositorio principal **CRAN** en la página web The R Project (<https://cran.r-project.org>).

### RStudio

Todo lenguaje de programación requiere de un IDE (*integrated development environment*), este es el programa capaz de cargar el lenguaje, comprenderlo y ejecutar sus instrucciones. A su vez, nos permite desarrollarlo y ofrece facilidades como el autocompletado o la clausura de paréntesis y corchetes automática.



El IDE más usado para R es **RStudio**, este tiene una versión limitada libre con descarga e instrucciones de instalación en la página de RStudio (<https://www.rstudio.com>).

La versión que usaremos individualmente en los ordenadores es la **desktop**. Es importante instalar antes el lenguaje.

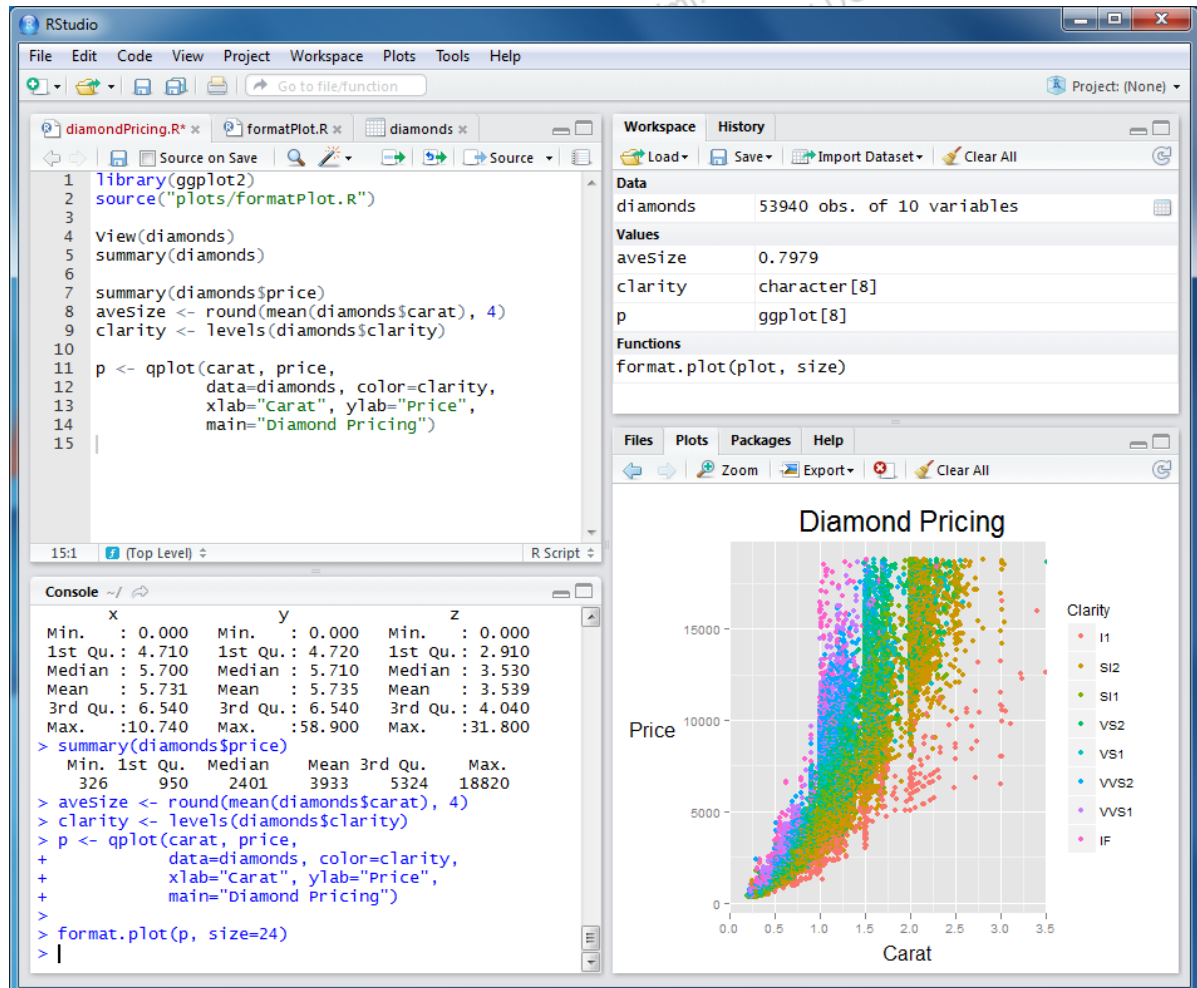


Imagen 1.3. RStudio.

Fuente: elaboración propia en R.

La pantalla consta de cuatro zonas que describimos de izquierda a derecha y arriba abajo:

- **Zona de escritura de script:** para escribir el código fuente que se ejecuta.
- **Zona con workspace y history:** para inspeccionar las variables definidas y ver el historial de comandos.
- **Zona con el terminal:** cuadrante en el que se ejecuta el código y donde aparecen los resultados en forma de caracteres.
- **Zona con files, plots, packages y help:** sección donde se despliega la ayuda pedida, se representan las imágenes, se exploran archivos o se consultan los paquetes instalados e instalables.



El formato en el que se trabaja a lo largo del curso es el **R-notebook**, que integra el código con los resultados de las ejecuciones.

Se puede obtener más información en la página sobre notebooks de Rstudio ([http://rmarkdown.rstudio.com/r\\_notebooks.html](http://rmarkdown.rstudio.com/r_notebooks.html)). Estos se pueden visualizar en pdf o en el mismo R-studio.

Los notebooks se encuentran en formato .rmd y .html. Se recomienda abrirlos en formato .html porque su visualización es mejor.

### Notebook con las operaciones fundamentales

Expresión de la operación en R	Significado
+	suma
-	substracción
/	división
^	potencia
%%	módulo
% / %	división de enteros
<	menor que
>	mayor que
<=	menor o igual que
>=	mayor o igual que
==	igual
!=	diferente
!x	NO lógico
x & y	Y lógico
x   y	O lógico
xor(x,y)	O exclusivo



**Descarga: Consulta el notebook UD1 N01**

Descárgate el archivo UD1 N01 y ejecútalo en R. También puedes verlo en .html

## V. Comandos esenciales en R y tipos de dato

En R hay unas 9000 librerías de características especializadas. La mayoría son librerías aportadas por la comunidad para poder realizar tareas específicas sobre datos, habitualmente con un alto nivel matemático/estadístico de sostén. Estas, a su vez, suelen contener datos incrustados como ejemplo de aplicación.

### Carga de librerías

Un resumen esencial de **carga de librerías** sería:

- *install.packages()*: visualiza los paquetes de datos disponibles en Internet.
- *install.packages(name)*: descarga e instala el paquete indicado.
- *library()*: visualiza los paquetes disponibles.
- *library(name)*: carga el paquete indicado.
- *data()*: visualiza los datos disponibles.
- *data(name)*: carga en memoria el dato indicado.

### Manejo de datos

Y un resumen de **manejo de datos**:

- *class(data)*: muestra el tipo de objeto.
- *dim(data)*: muestra las dimensiones del objeto.
- *ncol(data), nrow(data)*: muestra el número de columnas/filas del objeto.
- *names(data)*: muestra los nombres de las columnas.
- *objects(), ls()*: visualiza las variables cargadas en memoria.
- *rm(data1, data2)*: elimina las variables indicadas.
- *help(data), ?data*: muestra la ayuda asociada con el comando o variable.
- CTRL+L: borra la pantalla.

### Tipos de variables

Los tipos de variables que se usan en análisis de datos son los que enumeramos y definimos a continuación. Al programar, es crucial tener claro qué tipo de dato supone cada variable para saber los efectos de aplicación de las funciones del lenguaje sobre los mismos.

Los tipos de datos esenciales en R son:

#### Vector

Un **vector** es una variable en el significado comúnmente asumido.

#### Factor

Un **factor** es una variable categórica.

**Array**

Un **array** (arreglo) es una tabla de dimensión  $k$ .

**Matriz**

Una **matriz** es un caso particular de un array donde  $k = 2$ .

**data.frame**

Un **data.frame** (marco o base de datos) es una tabla compuesta de uno o más vectores y/o factores de la misma longitud pero que pueden ser de diferentes tipos.

**ts**

Un **ts** es una serie temporal y como tal contiene atributos adicionales como son frecuencia y fechas.

**Lista**

Una **lista** es una cadena con diferentes tipos de datos.

El tipo de elemento que contiene cada variable puede ser distinto. Hay variables **homogéneas**, en las que todos sus elementos son de un único tipo, y variables **heterogéneas**, que admiten diferentes tipos en sus elementos.

Se pueden convertir los tipos de los elementos de una variable con comandos como los siguientes:

- *as.numeric*: transforma en numérico un vector.
- *as.logical*: transforma en vectores con TRUE Y FALSE.
- *as.character*: transforma en strings o cadenas de letras.

Los dataframes son tablas que contienen información por columnas, siendo cada columna un atributo con un tipo diferente.

	X	Y	Z	costa
CA	1	0	1.000000	C
SE	2	3	2.500000	I
MA	3	2	1.666667	I
BA	4	5	2.250000	C
VA	5	9	2.800000	C

**Imagen 1.4.** Dataframe. *Fuente:* elaboración propia.

Las filas representan instancias o elementos y las columnas los distintos atributos. Se extraerá información de estas tablas y se analizarán los datos.





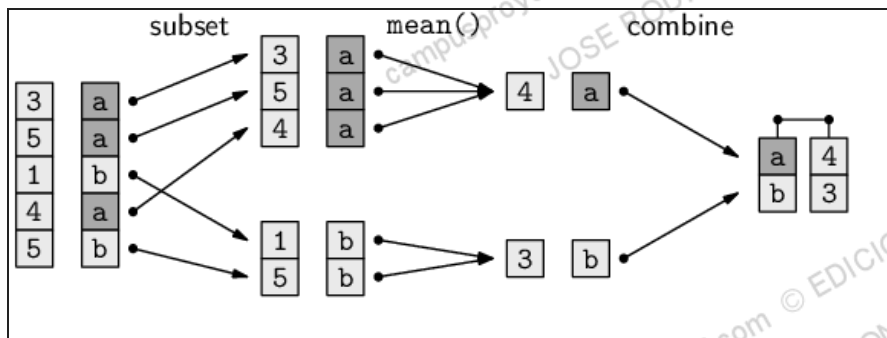
### Descarga: Consulta el notebook UD1 N02

Descárgate el archivo [UD1 N02](#) y ejecútalo en R. También puedes verlo en [.html](#)

## VI. Operaciones útiles sobre tablas, carga y descarga de datos

Cuando se manipulan tablas, lo que se pretende es aplicar funciones de agregación por grupos. Por ejemplo, se necesita hacer la media de altura de personas con distintos colores de ojos para compararlas.

Para calcular estos estadísticos agregados de una columna, existen las funciones *apply*, *tapply* y *sapply*, además, hay una librería completa (**dplyr**) para hacer operaciones de **división, agregación y combinación**.



**Imagen 1.5.** División, agregación y combinación. *Fuente:* Python for Social Science de Jean Mark Gawron.

### Estadísticos de agregación

- *sum(x)*: suma de los elementos de *x*.
- *prod(x)*: producto de los elementos de *x*.
- *max(x)*: valor máximo en el objeto *x*.
- *min(x)*: valor mínimo en el objeto *x*.
- *which.max(x)*: devuelve el índice del elemento máximo de *x*.
- *which.min(x)*: devuelve el índice del elemento mínimo de *x*.
- *range(x)*: rango de *x*.
- *length(x)*: longitud de *x*.
- *mean(x)*: media de *x*.
- *median(x)*: mediana de *x*.
- *var(x)*: varianza de *x*.
- *cor(x,y)*: correlación entre los valores de *x* e *y*.

El formato de trabajo usual para la carga de tabla es el *.csv* (*comma separated values*). R incorpora múltiples utilidades para la lectura y escritura de dataframes en *.csv*.

### Configuración en la carga y descarga elementos

Se pueden configurar en la carga y descarga elementos como:

- Separadores.
- Comas de decimales y de miles.
- Símbolos extraordinarios, valores asociados a valores faltantes.
- Encabezamientos.
- Fechas con diversos formatos.



#### Descarga: Consulta el notebook UD1 N03

Descárgate el archivo [UD1 N03](#) en R y este [csv](#). También puedes verlo en [.html](#)

## VII. Funciones y bucles

Las funciones en R son operadores que se definen como

- *entrada -> operaciones -> salida*

El formato de código es, como puede verse en el ejemplo siguiente, aquel en el que la función acepta dos argumentos, x e y, y devuelve su suma.

```
func <- function(x,y){
  #operaciones
  #respuesta
  return(x+y)
}
```

```
func(3,2)
```

**Imagen 1.6.** Función en R.

*Fuente:* elaboración propia.

Los bucles (**for**) y bifurcaciones (**if**) son las herramientas esenciales para diseñar flujo de trabajo en un lenguaje de programación.

**for**

En el **for**, se realizan una serie de operaciones secuencialmente mientras exista una condición:

```
for (val in sequence)
{
  statement
}
```

**Imagen 1.7.** Bucle for.

*Fuente:* elaboración propia.

**if**

En el **if**, se bifurca entre dos posibles operaciones según una condición:

```
if (test_expression) {
  statement1
} else {
  statement2
}
```

**Imagen 1.8.** Bifurcación if.

*Fuente:* elaboración propia.



**Descarga: Consulta el notebook UD1 N04**

Descárgate el archivo [UD1 N04](#) y ejecútalo en R. También puedes verlo en [.html](#)

## VIII. Resumen final



Se han introducido las herramientas necesarias para hacer tratamiento y carga de datos, así como el lenguaje R y su IDE Rstudio.

Por otra parte, se han dado los elementos para crear funciones y una base mínima de programación con bifurcaciones y bucles en R. Además, se han expuesto una serie de ejemplos sobre datos y casos concretos en los notebooks para asimilar a través de la experiencia lo expuesto.

No obstante, aprender un lenguaje nuevo suele ser un proceso lento y paciente.

## Ejercicios

### Caso práctico

Como repaso del tema y preparación para el Caso práctico final, se presenta el siguiente caso práctico.



Descárgate el archivo ACTIVIDAD1\_UD1 en R y la csv del caso. También puedes verlo en .html

Cuando lo hayas realizado, puedes descargar su solución y comprobar tus resultados.

### Solución



En los siguientes archivos dispones de la solución de la actividad propuesta:

- Solución en .html
- Solución en R

## Recursos

### Enlaces de Interés



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

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

The R project



<https://www.rstudio.com>

<https://www.rstudio.com>

RStudio



[http://rmarkdown.rstudio.com/r\\_notebooks.html](http://rmarkdown.rstudio.com/r_notebooks.html)

[http://rmarkdown.rstudio.com/r\\_notebooks.html](http://rmarkdown.rstudio.com/r_notebooks.html)

R Notebooks

### Glosario.

- **Bucle o bifurcación:** Partículas de programación que permiten diseñar programas al aportar itinerarios lógicos secuenciables o disyuntivos.
- **DataFrame:** Tabla con datos en los que cada columna representa un atributo y cada fila una instancia o entrada de los mismos. Las columnas pueden ser de naturaleza variada.
- **Dataset:** Conjunto de información en forma de datos sobre un asunto concreto que sirve para el estudio de ciertas variables objetivo, que constan de otras variables que actúan como referencias o predictores.
- **Librería o módulo:** Conjunto de funciones con aplicaciones en torno a un fin definido. Pueden tener dependencias entre sí.
- **R:** Lenguaje de programación de código abierto orientado a la estadística y tratamiento de datos.
- **RStudio:** IDE (entorno de desarrollo interactivo), para programar en R. Consta de distintas ventanas cubriendo las diversas necesidades de desarrollo.