



<b>R</b>	Lenguaje de programación creado por Ross Ihaka y Robert Gentleman, pensado para la manipulación, análisis estadístico y visualización de datos.
<b>Archivos con extensión .R</b>	Con él podemos editar en el Editor de Código de RStudio o ejecutar desde consola.
<b>Archivos con extensión .RData</b>	Almacena todos los objetos que están cargadas en memoria.
<b>Archivos con extensión .Rmd</b>	Almacena un proyecto de R Markdown.
<b>Archivos con extensión .Rproj</b>	Proyectos de RStudio que engloban un directorio de trabajo con su espacio de trabajo, historial y documentos de origen.
<b>Array</b>	Vector multidimensional o una matriz multidimensionales donde todos los elementos son del mismo tipo.
<b>Consola</b>	Muestra los resultados de todas las operaciones ejecutadas.
<b>Dataframe</b>	Similar a una matriz, puede contener datos elementos de distinto tipo.
<b>data.frame ()</b>	Función para construir un dataframe.
<b>cbind()</b>	Función para combinar dataframe a través de columnas.
<b>dim()</b>	Devuelve la dimensión del dataframe (filas por columnas).
<b>head()</b>	Función que muestra la cabecera (primeros 6 registros) del dataframe.
<b>ncol()</b>	Devuelve el número de columnas (variables) del dataframe.
<b>nrow()</b>	Devuelve el número de filas (observaciones) del dataframe.
<b>rbind()</b>	Función para combinar dataframe a traves de filas.
<b>str()</b>	Devuelve la estructura interna de cualquier objeto en R.
<b>str()</b>	Devuelve la estructura del dataframe.
<b>tail()</b>	Función que muestra los últimos 6 registros del dataframe.
<b>Date</b>	Clase de objeto para trabajar con fechas.
<b>as.character()</b>	Función para convertir variables de clase Date en variables de clase character.



<code>format()</code>	Función para convertir variables de clase Date en variables de clase character.
<code>diff()</code>	Calcula la diferencia en días, entre los términos sucesivos de un vector de fechas declaradas con la clase Date.
<code>unclass()</code>	Muestra la representación interna de un objeto en R.
<code>Sys.Date()</code>	Lee la fecha actual del reloj de nuestro ordenador y la codifica como un objeto de la clase Date.
<code>as.Date()</code>	Convierte un dato de fecha en modo carácter y a la clase Date. Con el argumento <i>format</i> podemos especificar de qué forma ha sido introducida la fecha.
<code>difftime()</code>	Calcula diferencia de fechas.
<b>Directorio de trabajo</b>	Lugar de donde se leen y guardan los archivos que utilizemos en la sesión.
<code>getwd()</code> get working directory	Comando para obtener cuál es el directorio de trabajo actual
<code>setwd()</code> <b>Editor de código</b>	Función utilizada para cambiar el directorio de trabajo. Aquí se escribe, se edita o pega el script para luego ser ejecutado.
<b>Environment (Entorno)</b>	Muestra todas las variables que tenemos cargadas en memoria.
<b>Estructuras de control</b>	Permiten controlar el flujo de ejecución de nuestra secuencia de comandos.
<code>break()</code>	Produce la salida inmediata del bucle.
Bucle	Composición compuesta que se repite n veces hasta alcanzar un límite definido.
for	Comando para implementar bucles. for( i in secuencia ){ expresión }
if	Para condicionar la ejecución u omisión de una instrucción. La condición debe ser una expresión del tipo booleana y siempre debe ir dentro de un paréntesis. A continuación, debemos escribir el proceso (instrucción compuesta) que queremos que ocurra en caso de que la sentencia utilizada como condición sea verdadera. Agregamos un else, y finalmente el proceso a utilizarse cuando la sentencia es falsa. if( condición ){ proceso 1 } else { proceso 2 }
<code>next()</code>	Si la condición es verdadera : if( condición ){ proceso } Produce un salto inmediato a la siguiente iteración.



repeat()	Comando para realizar bucles que repite un conjunto de instrucciones hasta que se satisfaga un criterio de parada. repeat { expresión if ( condición ) break }
while	Comando que realiza un ciclo pero no conocemos el número de ciclos que se precisan. while( condición ){ expresión }
<b>Función</b>	Toma datos de entrada y los usa para computar un resultado/producto. Tienen un nombre que utilizaremos para llamarla y entre paréntesis escribiremos los argumentos o parámetros que la función necesita separados por comas.
arrange()	Reordena filas de un conjunto de datos.
mutate()	Añade nuevas variables/columnas o transforma variables existentes.
read.table(), read.csv() y read.csv2() aggregate()	Permite importar y leer archivos.csv  Divide los datos en subconjuntos, calcula estadísticas de resumen para cada subconjunto y devuelve el resultado en un grupo por formulario.
as.character()	Convierte a tipo carácter.
as.factor()	Convierte a tipo factor.
as.integer()	Convierte a tipo entero.
as.logical()	Convierte a tipo lógico o booleano.
as.numeric()	Convierte a tipo numérico.
as.ordered()	Convierte a tipo factor ordenado.
class()	Función para conocer la clase de objeto.
addmargins()	Agrega las sumatorias marginales, de fila y columna a una tabla de contingencia.
cor()	Devuelve el coeficiente de correlación.
cor.test()	Pruebe la asociación entre muestras pareadas, utilizando uno de los coeficientes de correlación.
prop.table()	Tabla de contingencia con valores porcentuales.
round()	Función para redondear números.



<code>shapiro.test()</code>	Función para calcular la prueba de normalidad.
<code>t.test()</code>	Función para prueba de la media de una población normal.
<code>table()</code>	Función para crear tabla de contingencia.
<code>var.test()</code>	Prueba F para comparar dos varianzas.
<code>filter()</code>	Devuelve un conjunto de filas según una o varias condiciones lógicas.
<code>group_by()</code>	Agrupar un conjunto de filas seleccionado, en un conjunto de filas de resumen de acuerdo con los valores de una o más columnas o expresiones.
<code>ordered()</code>	Ordena una variable categórica.
<code>print()</code> <code>read.table()</code>	Imprimimos el resultado en la consola. Permite leer tablas en general.
<code>read.table(x,dec... )</code>	El character que se utilizó como decimal, suele ser el punto o la coma.
<code>read.table(x, fill...)</code>	TRUE/FALSE. Para indicar si se deben incorporar las filas incompletas. Por defecto es TRUE.
<code>read.table(x, header...)</code>	TRUE/FALSE Sirve para indicar si la tabla comienza con los nombres de las columnas.
<code>read.table(x, na.strings)</code>	Cómo se indican los datos faltantes en la tabla, por default es "NA"
<code>read.table(x, nrow...)</code>	Por defecto es -1, porque asume que la cabecera ocupa solo una fila.
<code>row.names()</code>	Puede ser un vector que dé los nombres de las filas, o un número que dé la columna de la tabla que contiene los nombres de las filas, o un string de caracteres que dé el nombre de la columna de la tabla que contiene los nombres de las filas.
<code>read_excel(), read_xls() y read_xlxs()</code> <code>col_names</code>	Permite importar y leer archivos .xls, .xlsx  TRUE para usar la primera fila como nombre de las columnas.
<code>read_excel(), read_xls() y read_xlxs() na</code>	Cómo se indican los datos faltantes en la tabla, por default es "" (vacío).



`read_excel()`, `read_xls()` y  
`read_xlxs()` `sheet`

`read_excel()`, `read_xls()` y  
`read_xlxs()` `skip`

En caso de querer leer otras pestañas que no sea la primera, debemos especificarlo.

En caso que tengamos una tabla cuya cabecera sea mayor a 1, debemos corregir este parámetro con números positivos.

`rename()`

Renombra variables en un conjunto de datos.

`rep()`

Función que permite crear un vector del largo que queramos repitiendo el número o caracter que le indiquemos.

`select()`

Devuelve un conjunto de columnas (variables).

`seq()`

Función para escribir una secuencia de números.

`set.seed()`

Función para generar datos aleatorios.

`subset()`

Genera otro dataframe seleccionando un subconjunto de datos a partir de uno original.

`summarise()`

Funciona de forma análoga a la función `mutate`, excepto que en lugar de añadir nuevas columnas crea un nuevo data frame.

`summarise()` / `summarize()`

Genera resúmenes estadísticos de diferentes variables en el conjunto de datos.

`tapply()`

Función que hace los cálculos por columna del dataframe y por las categorías que defina el usuario, dentro del grupo de observaciones.

`as_tibble()`

Función que sirve para forzar los objetos `data.frame` a `tbl_df`.

`tibbles` (`tbl`)

Similares a `data.frames` pero presentan ciertas mejoras que facilitan su manejo.

**`write.table()`**

Genera un archivo con los datos almacenados en un objeto, típicamente un dataframe, aunque puede ser de cualquier otro (vector, matriz, etc)

**History (Historial)**

Vemos el historial de las órdenes ejecutadas en consola.

**`list()`**

Función para crear una lista.

**Lista**

Colección ordenada de otros objetos, llamados componentes.

`length()`

Devuelve el número de elementos de un vector o una lista.

`names()`

Función que devuelve un vector con los nombres de los componentes.



<b>Matriz</b>	Vector bidimensional donde todos los elementos son del mismo tipo.
<code>cbind()</code>	Añade una columna a la matriz en el orden que escribamos la función.
<code>colnames()</code>	Devuelve el nombre de las columnas de una matriz.
<code>dim()</code>	Permite crear una matriz a partir de un vector estableciendo su dimensión.
<code>dim()</code>	Devuelve las dimensiones de una matriz.
<code>dimnames()</code>	Devuelve el nombre de las dimensiones de una matriz.
<code>length()</code>	Devuelve el número total de elementos de una matriz.
<code>matrix()</code>	Función para crear una matriz.
<code>mode()</code>	Devuelve el tipo de datos de los elementos de una matriz.
<code>ncol</code>	Argumento que define el número de columnas de una matriz.
<code>nrow</code>	Argumento que define el número de filas de una matriz.
<code>rbind()</code>	Añade una fila a la matriz.
<code>rownames()</code>	Devuelve el nombre de las filas de una matriz.
<b>Paquete</b>	Colección de funciones R, datos y código compilado.
	Permite instalar un paquete( el nombre entre comillas).
<code>install.packages("package_name")</code>	
<code>installed.packages()</code>	Permite ver qué paquetes están instalados.
<code>library()</code>	Función que activa paquetes.
<code>remove.packages()</code>	Permite eliminar paquetes.
<code>update.packages()</code>	Permite actualizar paquetes.
<b>Paquete dplyr</b>	Contiene una colección de funciones para realizar operaciones comunes de manipulación de datos.
<b>Paquete forcats</b>	Es parte del ecosistema Tidyverse pensado para trabajar con factores.
<code>fct_c()</code>	Concatena factores combinando niveles.



<code>fct_collapse()</code>	Función que agrupa los niveles a grupos que serían las nuevas etiquetas del nivel.
<code>fct_count()</code>	Muestra listado de frecuencia de los niveles de la variable.
<code>fct_drop()</code>	Elimina los niveles que no se utilizan.
<code>fct_expand()</code>	Incorpora niveles a la lista de niveles de un factor.
<code>fct_explicit_na()</code>	Función que etiqueta valores NA para visualizarlos.
<code>fct_other()</code>	Agrupar en “otras/os”, varias categorías de una variable con poca frecuencia.
<code>fct_recode()</code>	Función que permite recodificar los niveles de la variable.
<code>fct_relevel()</code>	Función que ordena variable categórica.
<b>Paquete lubridate</b>	Dispone de diversas funciones que facilitan la extracción de componentes de un objeto fecha y hora de clase POSIXct.
<code>ceiling_date()</code>	Redondea la fecha por exceso en la unidad especificada .
<code>dmy_hms()</code>	Incorpora la fecha en el orden día-mes-año-horas-minutos-segundos sin importar los separadores a usar.
<code>floor_date()</code>	Redondea la fecha por defecto en la unidad especificada.
<code>now()</code>	Función que indica hora actual.
<code>round_date()</code>	Redondea la fecha en la unidad que especifiquemos.
<code>today()</code>	Función que indica día actual.
<code>ymd_h()</code>	Fecha en el orden año-mes-día-hora.
<b>Paquete Tidyverse</b>	Conjunto de paquetes diseñados para manipulación y estructuración de datos.
<b>Posixt</b>	La clase Date nos permitía trabajar con fechas.
<code>as.POSIXct()</code>	Función que convierte un tipo de dato a la clase POSIXct.
<code>as.Posixlt()</code>	Para crear una fecha de tipo Posixlt.
<b>POSIXct</b>	Subclase Posixt que almacena internamente esta cifra como un número entero.



<b>POSIXlt</b>	Subclase Posixt que descompone en una lista con elementos para los segundos, minutos, horas, día, mes y año.
<b>R Studio</b>	Es un software open source . Entorno de desarrollo integrado que nos permite utilizar con facilidad el lenguaje R.
<b>Script</b>	Archivo que contiene un listado secuencial de líneas de código.
<b>Tabla de contingencia</b>	<p>Tabla que cuenta las observaciones por múltiples variables categóricas.</p> <p>Las filas y columnas de las tablas corresponden a estas variables categóricas.</p>
<b>Vector</b>	Conjunto de valores (números o símbolos), todos del mismo tipo ordenados de la forma (elemento 1, elemento 2, ... , elemento n), siendo n la longitud o tamaño del vector.