



01. R y R-Studio

Temas:

- * Conociendo R. Ventajas de incluir R en la tarea diaria. Distribución e instalación de R.
- * Primeros pasos con RStudio. Instalando RStudio- Uso de la ayuda

El objetivo de esta guía es describir brevemente qué es R, por qué utilizamos RStudio y cómo nos movemos dentro de este programa.

01.01. El lenguaje R

R es un lenguaje de programación creado por Ross Ihaka y Robert Gentleman, pensado para la manipulación, análisis estadístico y visualización de datos.

Con R todas son ventajas.

Por ser un lenguaje de código abierto, tiene la particularidad de contar con una extensa comunidad activa de colaboradores que desarrollan nuevas funciones para múltiples aplicaciones. Se **distribuye gratuitamente** bajo los términos de la [GNU General Public Licence](#)) – GPL. Los usuarios tienen la libertad de usar, estudiar, compartir (copiar) y modificar el software. Esto hace que existan actualizaciones permanentes de R con una gran compatibilidad entre las versiones (aunque no siempre es así).

R **permite guardar o exportar** los análisis estadísticos y los gráficos en varios formatos (jpg, png, bmp, ps, pdf, emf, pictex, xfig). A su vez, **permite automatizar procesos**, como, por ejemplo, al programar bucles (*loops*, en inglés) para analizar conjuntos sucesivos de datos.



01.02. ¿Por qué aprender R es fácil?

R es un lenguaje de programación interpretado (como Java) y no compilado (como C, C++, Fortran, Pascal, etc.), lo cual significa que los comandos escritos en el teclado son ejecutados directamente, sin necesidad de construir ejecutables. Por lo tanto, con escribir una simple cuenta en la consola ya tenemos un resultado por parte de R.

A su vez, R es un lenguaje simple y flexible. Por ejemplo, si queremos calcular una media usamos la función `mean()` (*mean*, es media en inglés) o para obtener un máximo, usamos `max()`. Como podrán imaginar, esto facilita muchísimo la escritura del código porque se vuelve intuitivo y fácil de recordar.

01.03. Lenguaje orientado a objetos

Para entender la lógica de la programación en R, es necesario que nos imaginemos que las variables, datos, funciones, resultados se guardan en la memoria activa de la computadora en forma de objetos con un nombre específico que le asignemos.

Podemos modificar o manipular estos objetos con operadores (aritméticos, lógicos, y comparativos) y funciones (que a su vez son objetos) para obtener el análisis que deseemos.

¡No nos vamos a adelantar ahora!

Volveremos en la próxima guía sobre este punto.

02. Instalando R



Podemos descargar R desde el Sitio oficial (<https://www.r-project.org/>), ya sea la última versión como las anteriores.



Si descargamos una nueva versión de R, la anterior no se borra. Esto es muy bueno, porque en caso que hagamos una actualización y algo no funcione, podemos volver fácilmente a la versión anterior.

Para instalar R en nuestra computadora:

Para Windows	https://cran.r-project.org/bin/windows/base/R-4.4.1-win.exe
Para Mac	R-4.4.1-arm64.pkg
Para Linux	https://cran.r-project.org/bin/linux/

En este curso trabajaremos con la última versión de R: versión R-4.4.1

03. Instalando R-Studio



RStudio es un software que nos permite utilizar con facilidad el lenguaje R. Se trata de un **entorno de desarrollo integrado** (IDE, de las siglas en inglés “*Integrated Development Environment*”). Esto es, un software que brinda distintas funcionalidades que vuelven más placentera la experiencia de escribir código. Utilizamos RStudio para que nuestro trabajo sea lo más parecido a escribir en un editor de texto, pudiendo contar con algo similar a un corrector ortográfico, un historial del trabajo realizado y permitiéndonos guardar el archivo a medida que avanzamos en él.



No es indispensable el uso de RStudio para trabajar con R, pero créannos que vuelve la tarea mucho más sencilla.

Entonces,

¿Qué es RStudio?

- Es un entorno de desarrollo muy práctico y potente para R



- Es un software open source (libre y gratuito bajo licencia GNU – GPLv3)
- Es multiplataforma: se puede instalar en Linux, Windows y Mac

¿Qué funcionalidades presenta RStudio?

- Selección de las versiones de R que tenemos instaladas para trabajar
- Creación de proyectos en donde concentramos los ficheros con los que estamos trabajando
- Gestión de archivos: ver los ficheros que tenemos en nuestra computadora
- Manejo de variables desde panel Entorno: ver las variables que creamos
- Descarga e instalación de paquetes por menú
- Historial de comandos: nos muestra todo lo que hemos ido haciendo en R
- Gestión de gráficos: podemos recurrir a gráficas anteriores y guardarlas como jpg, png o pdf
- Ayuda en línea

¿Por qué usamos RStudio?

- Respeta la filosofía tradicional de la consola de R
- Muestra los objetos cargados en memoria
- Permite ejecutar porciones de código con sólo marcarlos en los scripts
- Dispone de autocompletado de código
- Posee de un historial de comandos
- Tiene un visor de gráficos que unifica entornos
- Integra un gestor de paquetes instalados y/o cargados (librerías)

Para descargar RStudio vamos al sitio oficial:

<https://posit.co/download/rstudio-desktop/>

(versiones anteriores <https://www.rstudio.com>)

Windows 10	https://download1.rstudio.org/electron/windows/RStudio-2024.04.2-764.exe
------------	---

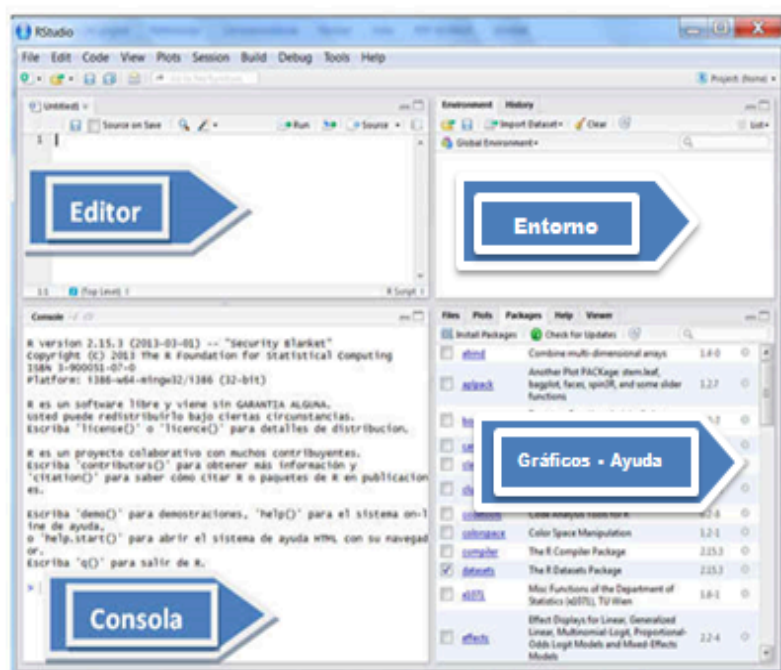
Mac OS X 12+ (64-bit)	https://download1.rstudio.org/electron/macOS/RStudio-2024.04.2-764.dmg
Ubuntu 20 Debian 11 (64-bit)	https://download1.rstudio.org/electron/focal/amd64/rstudio-2024.04.2-764-amd64.deb

Mas información: <https://rstudio-education.github.io/hopr/starting.html>

04. Reconociendo R-Studio

Cuando abrimos R-Studio por primera vez y creamos nuestro primer archivo, veremos que la pantalla se divide en 4 ventanas:

- 1- Editor de código
- 2- Entorno / Historial
- 3- Consola
- 4- Archivos / Gráficos / Paquetes / Ayudas



04.01. Ventana de edición del código



En la ventana de **edición del código** se escribe, se edita o pega el script para luego ser ejecutado.

Los iconos son:



Guarda el script creado en el editor

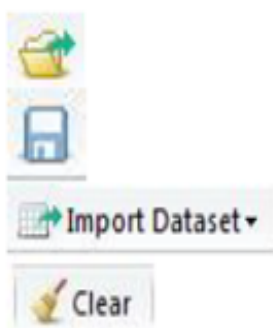
Busca y/o reemplaza palabras dentro del script

Autocompleta sintaxis de funciones

Ejecuta la línea o la porción de código seleccionada

En esta ventana y en la consola podemos usar el **Asistente de código**, una herramienta que permite autocompletar la función que vamos escribiendo desde el teclado. Para ejecutar el autocompletado debemos pulsar la tecla de tabulación (**Tab**) y nos aparecerá un menú desplegable donde se muestran las posibilidades de finalizar las funciones y el esquema de argumentos obligatorios que tiene asociado dicha función asociada.

04.02. La consola



En la **consola** es donde se muestran los resultados de todas las operaciones ejecutadas. También se pueden escribir allí en forma directa comandos de R y ejecutarlos con la tecla **Enter**.

Lo que escribimos en la consola no se guarda. Solo se guardan las cosas que escribamos en la ventana de edición de código.



La consola es una ventana directa al procesamiento con R. Si no usáramos R-Studio, trabajaríamos con una consola muy similar. En cada nueva línea siempre aparece un cursor a la derecha (**prompt**) que indica que podemos escribir. El prompt característico de R es `>`.

04.03. Entorno / Historial

En el recuadro superior izquierdo encontraremos las pestañas *Environment* (Entorno) e *History* (Historial). El **Entorno** nos muestra todas las variables que tenemos cargadas en memoria, mientras que en el panel **Historial** vemos el historial de las órdenes ejecutadas en consola.

Allí  encontramos los siguientes iconos:

Abre espacio de trabajo almacenado

Guarda el espacio de trabajo activo. Es decir, todos los objetos que tenemos cargados en la memoria en ese momento

Importa datos

Borra todos los objetos del espacio de trabajo / Borra todo el historial

Otra forma de acceder al historial de funciones es desde la consola. Si nos situamos en el prompt activo y pulsamos la tecla flecha hacia arriba, veremos la lista completa de comandos que ya hemos ejecutado desde el principio de la sesión de trabajo. Esto resulta un atajo útil cuando nuestra última línea ejecutada presenta problemas o queremos probar algo, pero no guardarlo.

04.05. Archivos – Gráficos – Paquetes – Ayuda – Visor

Este bloque se ubica en la parte inferior derecha de la pantalla y tiene acceso a los paneles *Files* (Archivos), *Plots* (Gráficos), *Packages* (Paquetes), *Help* (Ayuda) y *Viewer* (Visor).



Dentro de la pestaña **Packages** podemos gestionar las librerías de funciones (llamados en R, paquetes). Si contamos con acceso a internet, cuando busquemos algún paquete en particular se buscará entre los paquetes ya instalados en nuestra máquina y de no hallarse, lo hará en la red y se descargará e instalará de una manera muy sencilla.

La pestaña **Plots** nos muestra todos las gráficas que hemos realizado en nuestra sesión de trabajo. Desde aquí también podemos descargarlas en distintos formatos.

Cuando trabajamos dentro de un proyecto de R, la pestaña **Files** nos resulta muy útil, porque nos permite navegar dentro de los archivos que tenemos en nuestra computadora, sin la necesidad de cambiar de ventana.

Por último, en **Help** encontramos la ayuda y documentación de cada función y/o comando, como así también la de los paquetes seleccionados. Toda esta información se encuentra en el idioma original de la documentación que habitualmente es el inglés.

04.05. Tipos de archivos de RStudio

En RStudio podemos abrir archivos con distintas extensiones:

- Archivos con extensión **.R**: archivos de R que podemos editar en el Editor de Código de RStudio o ejecutar desde consola.
- Archivos con extensión **.RData**: almacenan todos los objetos que están cargadas en memoria. Es el mismo contenido del workspace que se visualiza en el panel *Environment*.
- Archivos con extensión **.Rhistory**: almacenan el historial de sentencias ejecutadas en RStudio que se visualizan en el *History*.
- Archivos con extensión **.Rproj**: proyectos de RStudio que engloban un directorio de trabajo con su espacio de trabajo, historial y documentos de origen.
- Archivos con extensión **.Rmd**: Almacena un proyecto de R Markdown

