

# Introducción a RStudio

M. A. Ibáñez

14/09/2023

## Índice

<b>1. ¿Qué es RStudio?</b>	<b>1</b>
<b>2. ¿Cómo Instalar RStudio?</b>	<b>1</b>
<b>3. Primeros pasos con RStudio</b>	<b>2</b>
3.1. Cargar e instalar paquetes en RStudio . . . . .	3
3.2. Proyectos en RStudio . . . . .	4
3.3. Archivos de instrucciones (script) en RStudio . . . . .	5

## 1. ¿Qué es RStudio?

RStudio es un entorno de desarrollo integrado (IDE) para R. Incluye la consola de R, un editor de instrucciones con resaltado de sintaxis que admite la ejecución directa de código, visualización del área de trabajo, gestión de la carpeta de trabajo y otras utilidades.

Rstudio tiene una página oficial en <https://www.rstudio.com/>.

## 2. ¿Cómo Instalar RStudio?

Para instalar Rstudio previamente tiene que estar instalado R.

Hay dos versiones de Rstudio: RStudio para escritorio (RStudio Desktop) y RStudio para servidores (Rstudio Server).

La primera es una versión de Rstudio para instalar y usar en ordenadores personales. La segunda es una versión para instalar en un servidor y dar servicio a múltiples usuarios.

Nosotros vamos a trabajar con la versión para escritorio.

Al igual que ocurre con R, RStudio puede ser instalado y ejecutado en distintos sistemas operativos: Windows, Mac y Linux.

En la página web de Rstudio hay una página de descargas [download](#) donde podemos seleccionar el instalador para nuestro sistema operativo.

En nuestro caso, RStudio Desktop para windows [RStudio-2022.07.1-554.exe](#).

Descargamos el instalador y lo guardamos en una carpeta del ordenador. Ejecutamos el archivo y seguimos los pasos de la instalación.

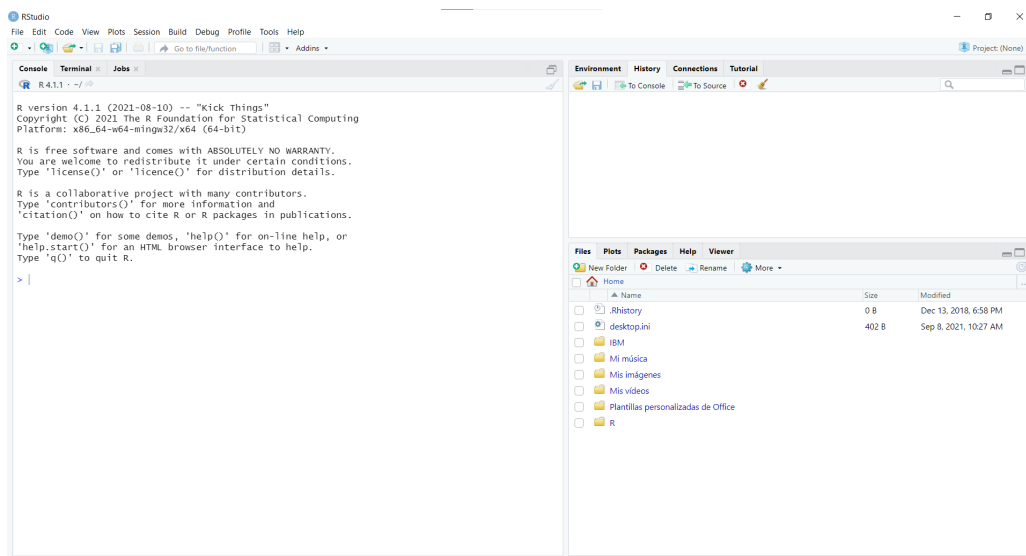


Figura 1: Interfaz de Rstudio

La instalación creará en el escritorio un icono de acceso directo a la aplicación para lanzar RStudio.

Si no se crea dicho acceso directo, pinchamos en el icono de inicio de windows y en el menú desplegable buscamos la carpeta RStudio.

Dentro de esta carpeta está el lanzador de la aplicación (que se llama Rstudio). Con el botón izquierdo del ratón, pulsamos sobre el lanzador y sin soltar el botón arrastramos el icono hasta el escritorio. De esta forma ya tenemos un acceso directo para lanzar RStudio.

### 3. Primeros pasos con RStudio

Cuando lanzamos Rstudio, inmediatamente se lanza R y se abre una ventana con el entorno de desarrollo de Rstudio.

En la figura 1 se muestra una imagen de la aplicación Rstudio.

Inicialmente, la ventana está dividida en tres ventanas o paneles. A la izquierda está el panel con la consola de R. Es la misma consola de R que hemos visto anteriormente. En ella podemos teclear y ejecutar código R.

En la parte superior derecha, aparece el panel **Environment** que muestra los objetos almacenados en el área de trabajo. Inicialmente estará vacía y a medida que vayamos creando objetos irán apareciendo en dicho panel.

En la parte inferior derecha, está el panel **Files** que muestra los archivos almacenados en la carpeta de trabajo, con una serie de iconos para crear nuevas carpeta, borrar archivo, renombrar archivos y otros similares.

En este panel existen distintas pestañas: **Plots** donde se mostrarán los gráficos que realicemos, **Packages** para instalar y cargar paquetes y **Help** para visualizar la ayuda.

### 3.1. Cargar e instalar paquetes en RStudio

En panel **Files** está la pestaña **Packages** que al pulsar sobre ella se muestra una lista con los paquetes instalados. La lista muestra el nombre, una descripción y la versión del paquete. Dispone de una barra lateral de navegación que facilita el desplazamiento por la lista.

En el lado izquierdo del nombre de los paquetes hay un cuadrado. Los paquetes que están cargados en memoria aparecen con dicho cuadrado marcado con una marca de verificación.

Si deseamos cargar en memoria un paquete, marcamos el cuadrado de dicho paquete pulsando sobre él con el botón izquierdo del ratón.

Vemos que en la consola de R se ha ejecutado la instrucción `library()` con el nombre del paquete que hemos cargado dentro del paréntesis, tal y como indicamos en tema de introducción a R.

Aunque esta es una forma de cargar en memoria paquetes, lo más correcto es utilizar la función `library()` en las primeras líneas del script y con esto cargar en memoria los paquetes que va a utilizar el script. De esta manera, cuando en el futuro volvamos a utilizar el script o queramos compartir dicho script con otra persona, sabremos que paquetes se van a utilizar.

En el panel **Packages** existe el icono **Install** para instalar paquetes. Al pulsar sobre él se abre una ventana (ver figura 2) donde podemos teclear el nombre del paquete que deseamos instalar.

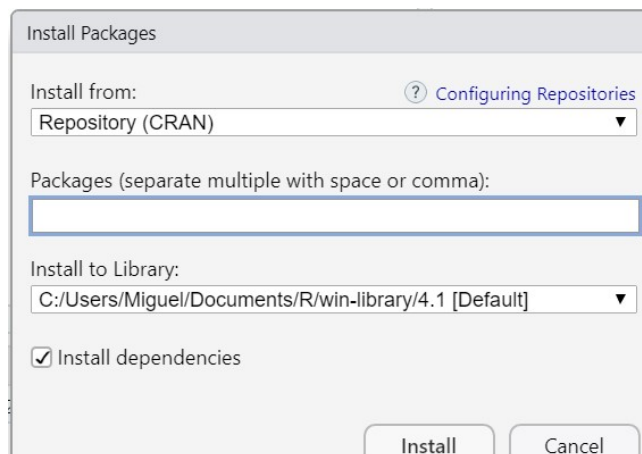


Figura 2: Instalar un nuevo paquete

Según tecleamos el nombre del paquete, se va desplegando una lista de paquetes disponibles en el CRAN y que comienzan con las letras tecleadas. Seleccionamos el paquete deseado y pulsamos **Install**.

En la consola de R aparecerá la instrucción con la función `install.packages()` y dentro del paréntesis el nombre del paquete seleccionado.

En ese momento comienza el proceso de instalación del paquete. La información del procesos aparece en la consola de R en letras rojas. Si el paquete depende de otros paquetes que no están instalados, estos paquetes se instalarán de forma automática.

Una vez instalado el paquete podemos comprobar que aparece en la lista de paquetes instalados. Ya solo queda cargarlo en memoria para su utilización.

### 3.2. Proyectos en RStudio

Para agilizar el manejo de carpetas de trabajos en R, Rstudio utiliza el concepto de **proyecto**. Un proyecto es una carpeta del ordenador donde están almacenado los archivos que vamos a utilizar en el análisis.

Podemos crear un nuevo proyecto pinchando en el menú **File** de RStudio y luego en el submenú **New Project**. Se abre una ventana (ver figura 3) en la que nos pide que indiquemos si queremos crear el proyecto creando una nueva carpeta o si queremos utilizar una carpeta ya existente para asociar con el proyecto.

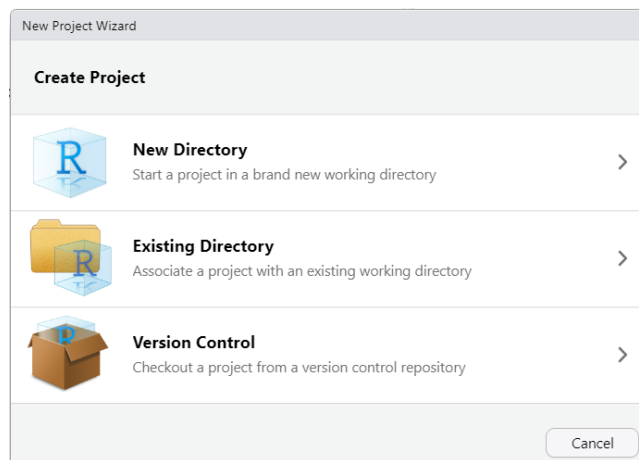


Figura 3: Crear un nuevo proyecto

Lo normal es seleccionar la segunda opción. Se abrirá una nueva ventana (ver figura 4) donde pinchando en **Browse...** nos permitirá navegar por el sistema de archivos de ordenador y seleccionar la carpeta deseada.

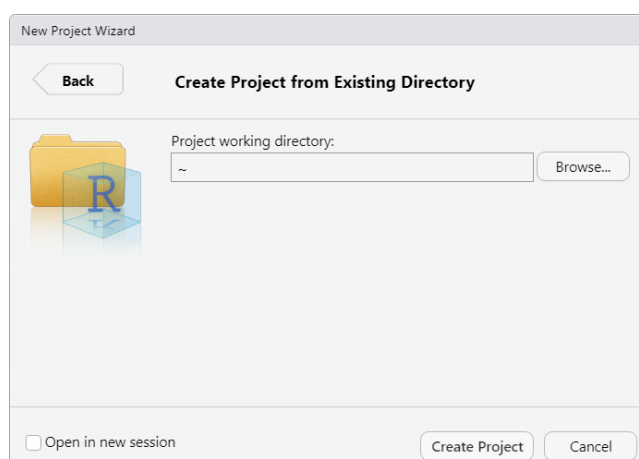


Figura 4: Navegar por el sistema de archivo para seleccionar la carpeta

Finalmente se pulsa en el botón **Create Project** y la aplicación se reinicia apareciendo en el panel **File** los archivos almacenados en la carpeta seleccionada. Desde este momento R considera dicha carpeta como la carpeta de trabajo.

Al crear el proyecto, se crea en la carpeta un archivo con el nombre de la carpeta y la extensión `.Rproj`. Los archivos con la extensión `.Rproj` están asociados a la aplicación Rstudio, de forma que pinchando con el botón izquierdo del ratón sobre dicho archivo se abre Rstudio con dicha carpeta como carpeta de trabajo.

En el menú **File** también existe la posibilidad de abrir un proyecto ya creado utilizando el submenú **Open Project...**

### 3.3. Archivos de instrucciones (script) en RStudio

Ya hemos indicado que la forma habitual de trabajar en R es mediante archivos de instrucciones.

En RStudio podemos crear un nuevo archivo de instrucciones accediendo al menú **File**, luego al submenú **New File** y por último seleccionado **R script** tal y como se muestra en la figura 5.

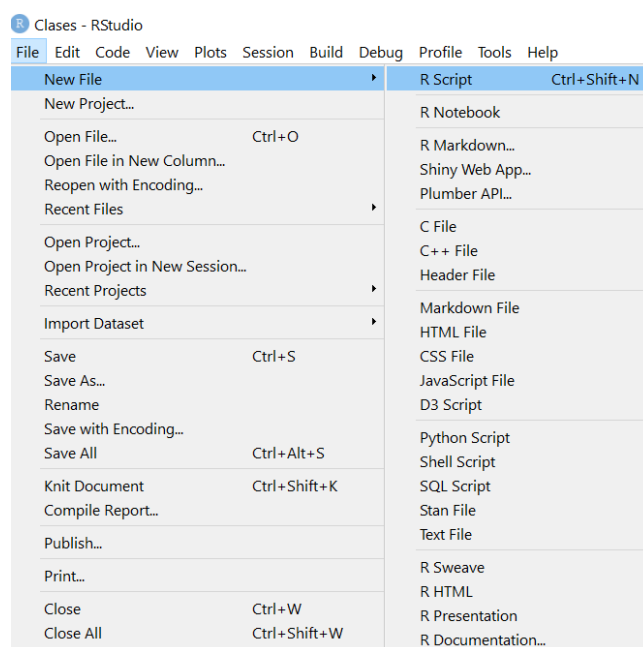


Figura 5: Crear un archivo nuevo de instrucciones

Inmediatamente se abre un nuevo panel, en la parte superior derecha, con un editor de instrucciones.

A título de ejemplo, podemos teclear la dos siguientes instrucciones:

```
> x <- c(34,45,67,23,12)
> mean(x)
[1] 36.2
```

Cada instrucción tiene que ser una línea en el archivo, tal y como se muestra en la figura 6.

Para ejecutar una línea de instrucciones, colocamos el curso en la línea y pulsamos sobre el icono **Run**. Inmediatamente la instrucción es enviada a la consola de R y ejecutada.

En este ejemplo, lo que veremos es que en el panel **Environment** aparece el objeto `x` que hemos creado con la instrucción ejecutada. En este caso, `x` es un vector numérico con 5 valores.

Una forma alternativa de ejecutar una línea de instrucciones, es utilizar la combinaciones de las teclas

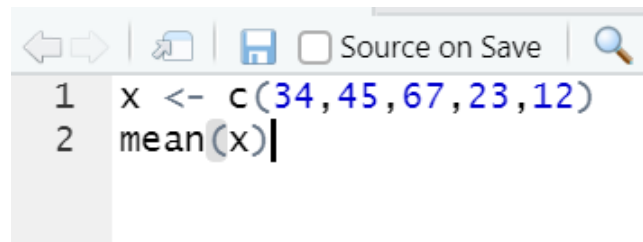


Figura 6: Editor de script

Ctrl y Enter.

La segunda instrucción utiliza la función `mean()` para calcular la media de los 5 valores almacenados en `x`, que es 36.2, tal y como se muestra en la consola cuando la ejecutamos.

Más adelante veremos este tipo de instrucciones.

También es posible ejecutar un conjunto de instrucciones seleccionándolas y a continuación pulsar sobre el icono **Run** o pulsar las teclas **Ctrl** más **Enter**.

Una vez hemos creado un archivo de instrucciones, podemos guardarlo en la carpeta de trabajo utilizando el menú **File** y luego el submenú **Save as**, tal y como se muestra en la figura 7

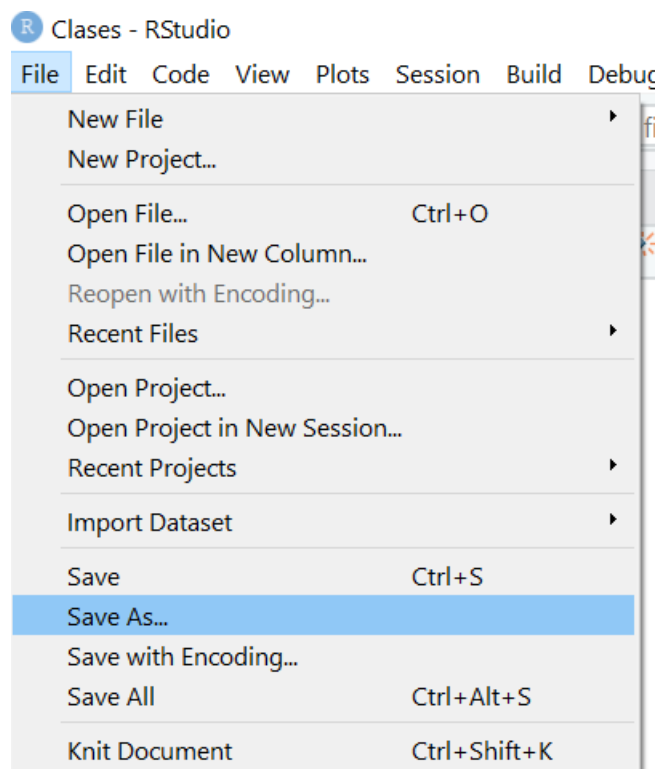


Figura 7: Guardar un archivo de instrucciones

La ventana emergente nos pide un nombre para el archivo. Los script tiene extensión `.r` por lo que tecleamos, por ejemplo, `MiPrimerCodigo.r` y pulsamos en **Save**.

Comprobamos como en el panel **Files** aparece el archivo creado.

En el menú **File** podemos abrir un script ya creado utilizando el submenú **Openfile** o guardar las modificaciones realizadas en un script con el submenú **Save**.

Debajo de la barra de menús hay una barra de herramientas con símbolos para realizar dicha tareas (ver figura 8)

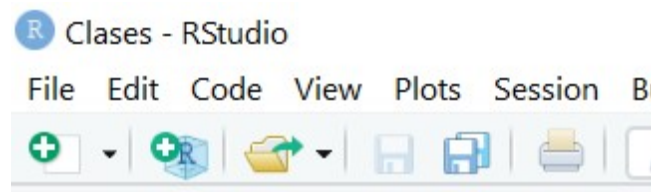


Figura 8: Barra de herramientas par manejar archivos

Una de las ventajas que tiene el editor de Rstudio es que tiene la función de auto completar, es decir, según tecleamos instrucciones nos proporciona opciones para completar la instrucción y en el caso de que sea una función, ayuda sobre su sintaxis, tal y como se muestra en la figura 9 donde se está tecleando la función `library()`.

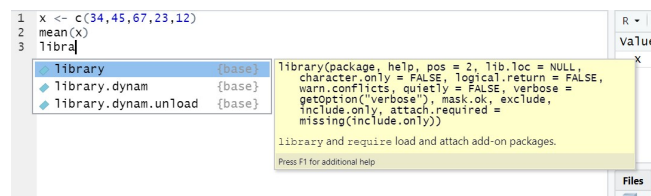


Figura 9: Propiedad de autocompletar del editor de Rstudio

El editor tiene también la función de resaltado de sintaxis con la que diferencia elementos del texto con distintos colores.

A partir de este momento utilizaremos el editor de instrucciones para crear y ejecutar nuestros propios script o cargar los script que utilizaremos en el resto de las clases.

Recordad que un archivo de instrucciones siempre ha de ejecutarse de forma secuencial, siguiendo el orden de instrucciones que tiene.