

Empezar a trabajar con RStudio

Sitio: [Plataforma de Formación On line del Instituto Andaluz de
Administración Pública](#)
Curso: (I22F-PT05) Entorno de Programación R
Libro: Empezar a trabajar con RStudio

Imprimido por: ALFONSO LUIS MONTEJO RAEZ
Día: viernes, 1 de abril de 2022, 08:31

Tabla de contenidos

1. Arrancando RStudio

- 1.1. Barra de Menús
- 1.2. Consola
- 1.3. Gestión de datos
- 1.4. Gestión espacio de trabajo

2. Operaciones habituales

- 2.1. Directorio de trabajo
- 2.2. Cargar y guardar datos
- 2.3. Historial
- 2.4. Ayuda

3. Gestión de paquetes

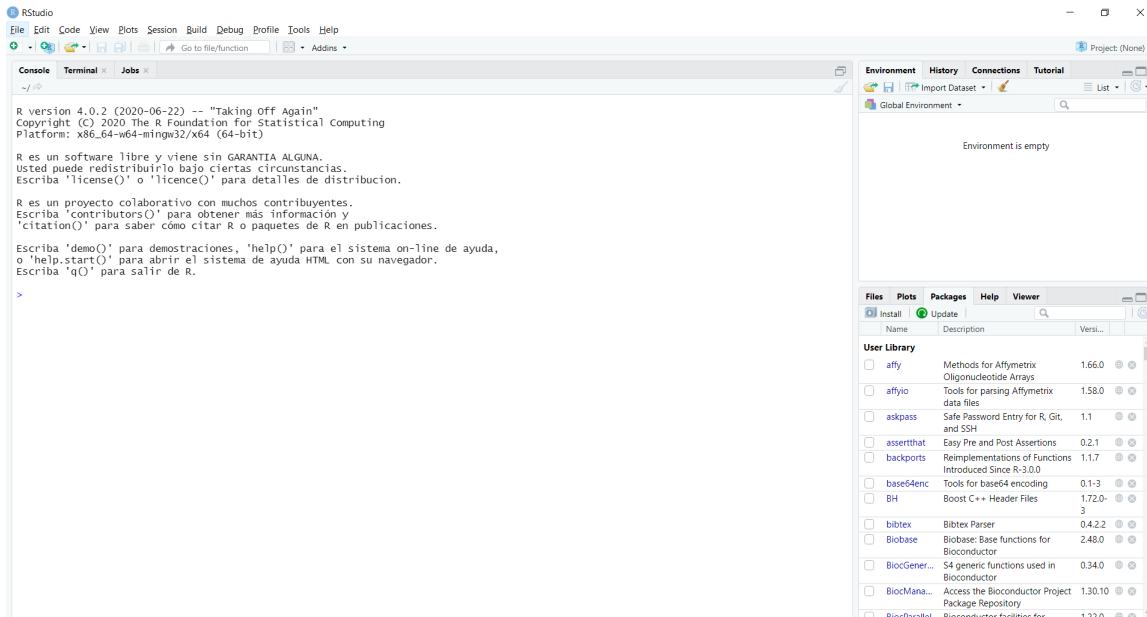
- 3.1. Paquetes interesantes
- 3.2. Operaciones sobre los paquetes
- 3.3. Instalar paquetes
- 3.4. Cargar paquetes
- 3.5. Actualizar paquetes

4. Formas de trabajo

- 4.1. Modo interactivo
- 4.2. Modo Script
- 4.3. Proyecto de R

1. Arrancando RStudio

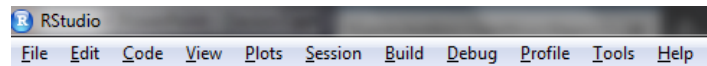
Una vez instalado el programa ya podemos arrancarlo y lo que nos encontramos es esto:



Cuando arrancamos podemos ver el área de trabajo de RStudio, que está dividida en tres partes. Ahora vamos a empezar describiendo el programa y cada una de las ventanas que nos encontramos.

1.1. Barra de Menús

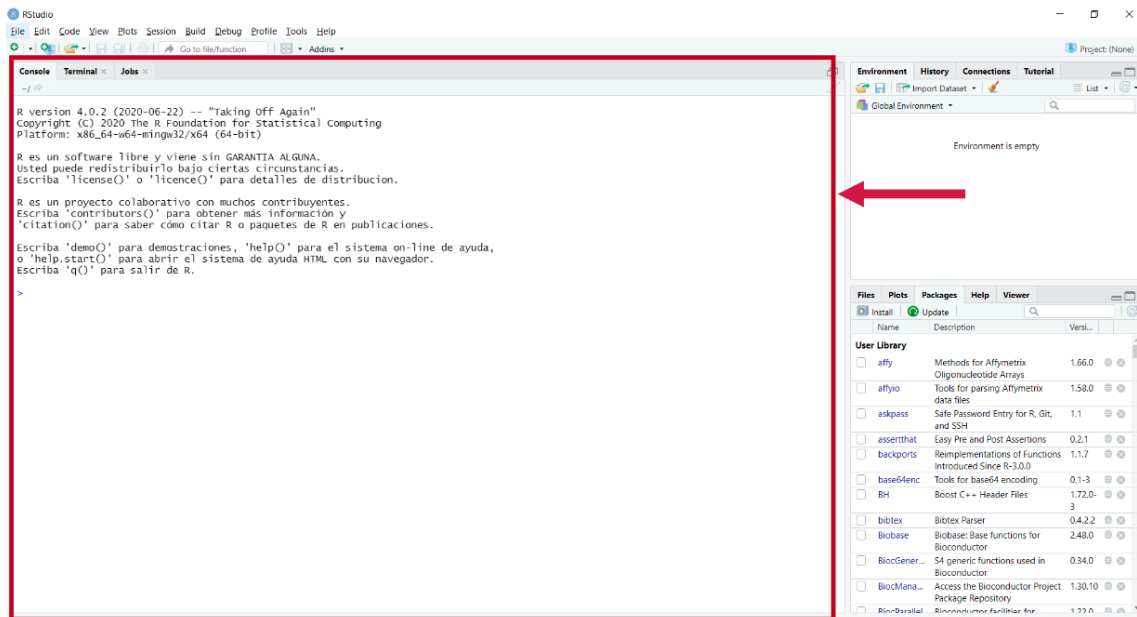
La barra de menús es donde se encuentran todas las acciones que se pueden hacer con RStudio.



- **File, Edit, View y Help.** Son menús similares a otros programas bajo Windows.
- **Plots.** Gestión de los gráficos generados durante el análisis.
- **Session.** Gestión del espacio de trabajo y de la sesión activa.
- **Code, Debug y Profile.** Gestión, depuración y auditoría del código.
- **Tools.** Gestión de paquetes y opciones de configuración del programa.
- **Build.** Creación de paquetes.

1.2. Consola

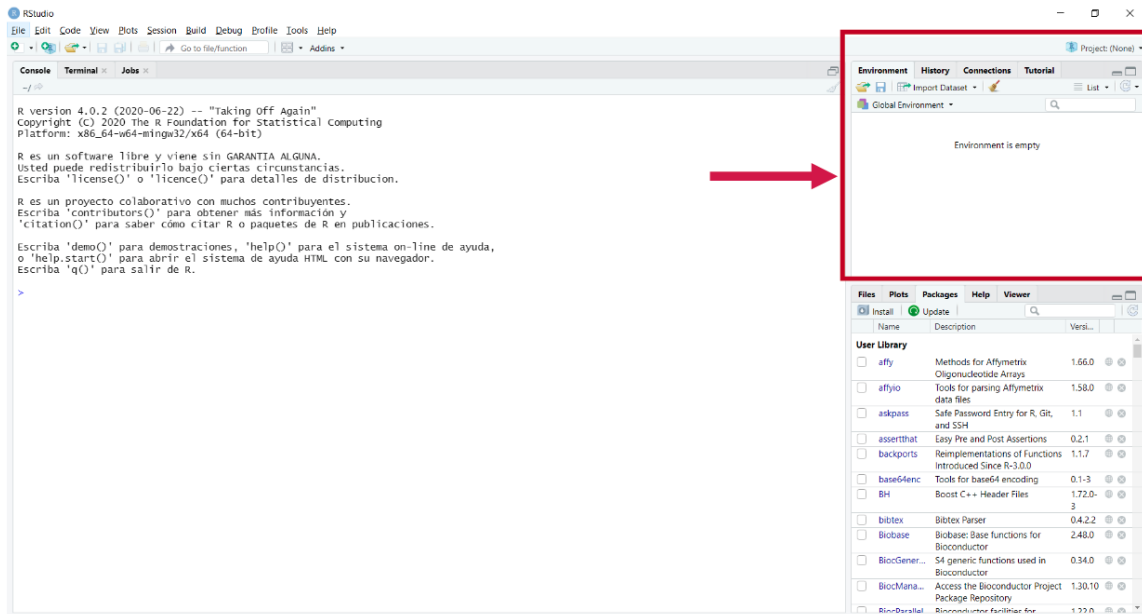
La consola es donde se dan las órdenes y donde irán saliendo los resultados. Es la misma consola que ya conocemos de R y se trabaja de la misma forma.



1.3. Gestión de datos

En esta pantalla podemos encontrar varias cosas importantes:

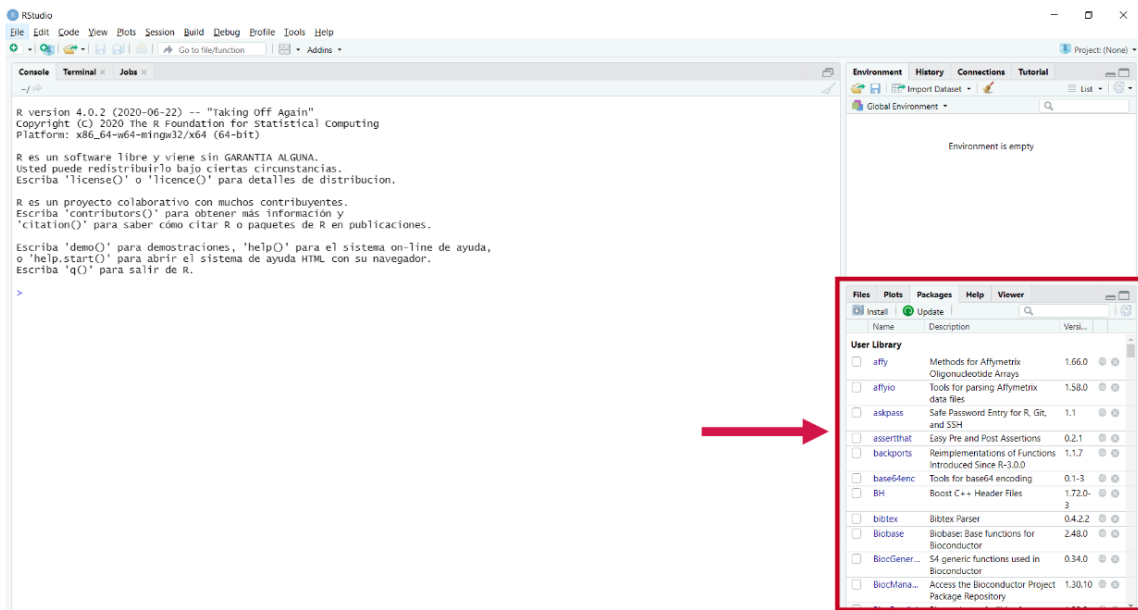
- **Environment** (Gestión de datos) donde irán apareciendo todos los objetos que vamos creando junto con una pequeña descripción del contenido.
- **History** (Historial) donde se almacenan todas las órdenes que hemos ido dando en la sesión de trabajo.



1.4. Gestión espacio de trabajo

En esta pantalla podemos encontrar varias cosas importantes:

- **Files** (Ficheros) donde podemos ver el directorio de trabajo.
- **Plots** (Gráficos) donde irán apareciendo los gráficos que vamos haciendo.
- **Packages** (Paquetes) donde podemos gestionar los paquetes o librerías (instalación, carga, actualización, etc.).
- **Help** (Ayuda) donde tenemos disponible toda la ayuda que ofrece R sobre las funciones que podemos usar.
- **Viewer** (Visor) donde aparecerán salidas gráficas como mapas, aplicaciones interactivas, etc.



2. Operaciones habituales

Ya que conocemos el programa RStudio vamos a ver las principales acciones que tenemos que tener en cuenta antes de empezar a trabajar como:

- Seleccionar el directorio de trabajo.
- Cargar y guardar datos.
- Historial.
- Ayuda.

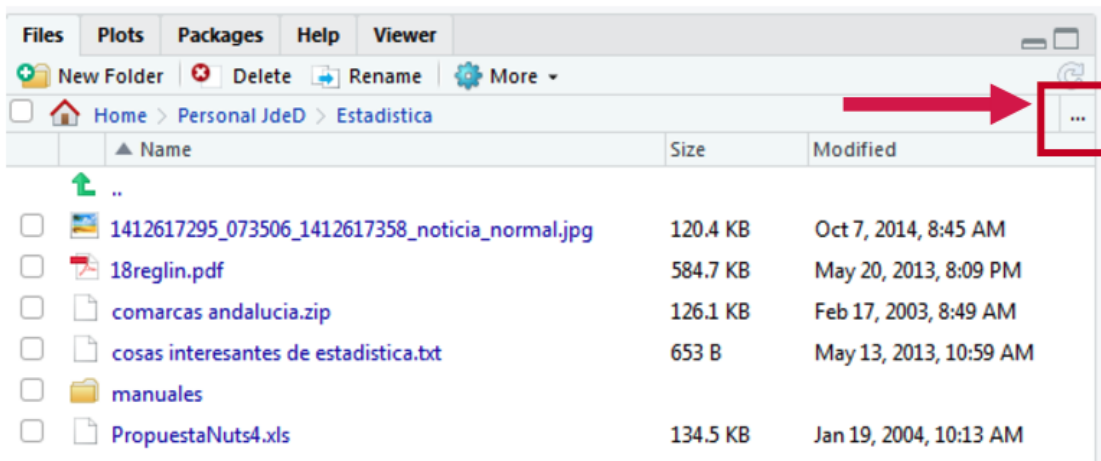
2.1. Directorio de trabajo

Como ya vimos en la parte de R, el directorio de trabajo es el **lugar donde se almacenan físicamente los objetos** que vamos creando. Normalmente será una carpeta de nuestro disco duro o algún dispositivo extraíble (pendrive o discos duros portátiles).

En definitiva, es el directorio en el que se va a desarrollar todo el trabajo en cada sesión.

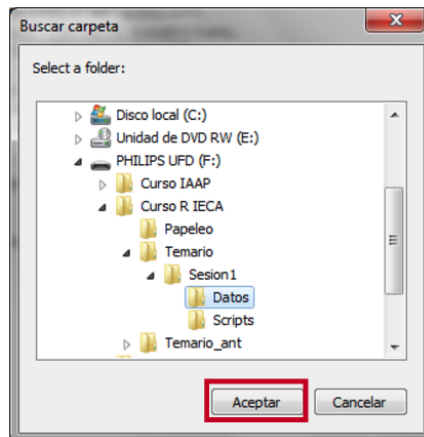
Para poder seleccionar/cambiar este directorio lo tenemos que hacer desde la pestaña **Files** (Ficheros), siguiendo los siguientes pasos:

1. Elegimos la pestaña **Files** (Ficheros)



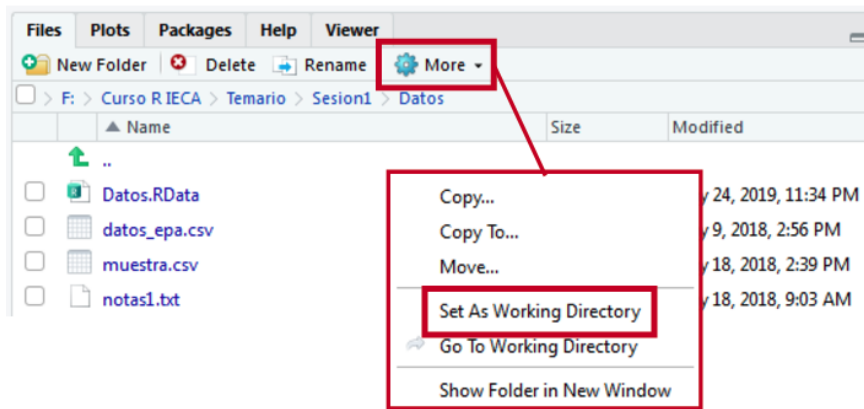
Lo que vemos en esta pantalla sería el contenido del directorio de trabajo actual. Para cambiar de directorio tenemos que pulsar sobre el botón de los tres puntitos.

2. Navegamos para seleccionar la nueva ubicación



Una vez localizada la carpeta donde queremos que esté el nuevo directorio de trabajo pulsamos **Aceptar**.

3. Convertimos la nueva ubicación en directorio de trabajo



Debemos pulsar el botón **More** y seleccionar la opción **Set As Working Directory**

Con esta acción conseguimos que la carpeta donde vamos a trabajar sea (en el ejemplo) un pendrive que está en la unidad F: de mi ordenador.

Todas estas acciones el programa las ha ido traduciendo a órdenes de R; en concreto ha creado la siguiente sentencia y la ha ejecutado automáticamente.

```
setwd("F:/Curso R IECA/Temario/Sesión1/Datos")
```

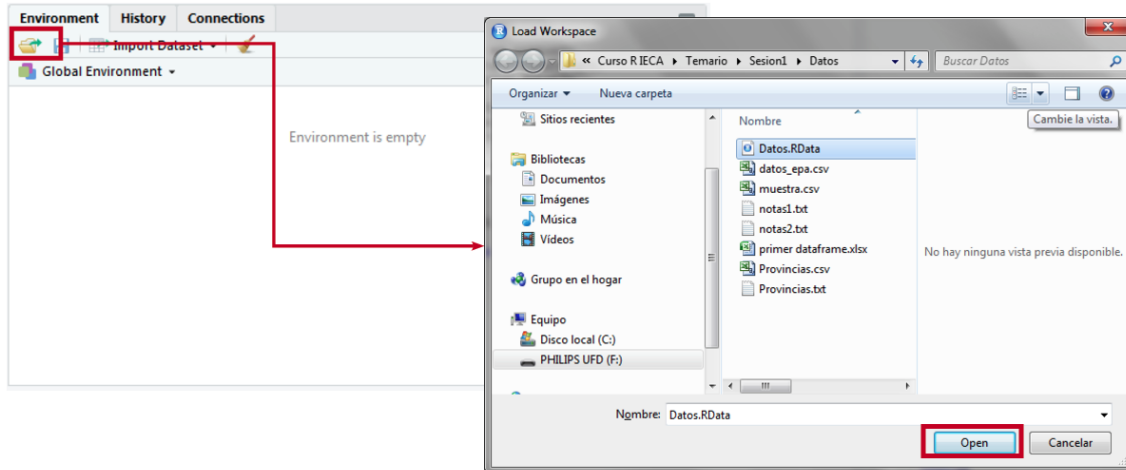
que, como ya sabemos, sirve para cambiar el directorio de trabajo.

2.2. Cargar y guardar datos

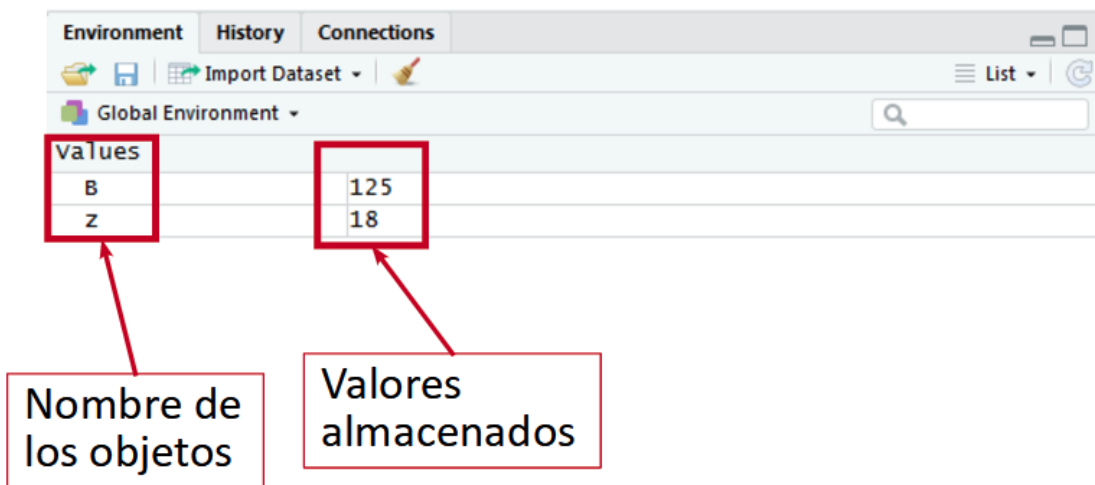
En este apartado podemos cargar un fichero de datos en formato RData, que ya tengamos creado, o crear un fichero para guardar todos los objetos.

Para cargar un fichero

Para abrirlo tenemos que irnos a la pestaña **Environment** (Entorno) y hacer lo siguiente:

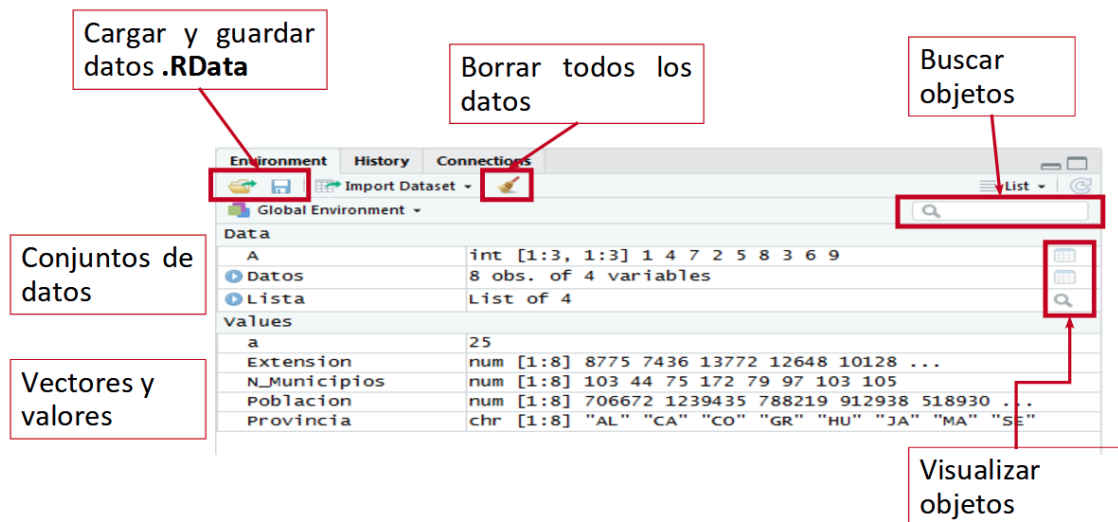


Pulsamos sobre el icono de la carpeta y buscamos la ubicación del fichero RData que queremos abrir y pulsamos **Open**.



En esta pestaña aparecen los objetos clasificados por sus tipos (vectores, data frames, etc); esto no os suena de nada pero en la siguiente unidad empezaremos a trabajar con ellos. Además también tenemos los nombres e información sobre el contenido de los objetos.

Una muestra de lo que tiene esta pestaña, cuando tenemos muchos tipos de objetos, sería esta:



La pestaña de **Environment** (Entorno) también nos permite hacer más cosas, como guardar todos los objetos, borrarlos, buscar un objeto, podemos visualizar los datos, etc. Todo esto lo iremos viendo en las próximas unidades.

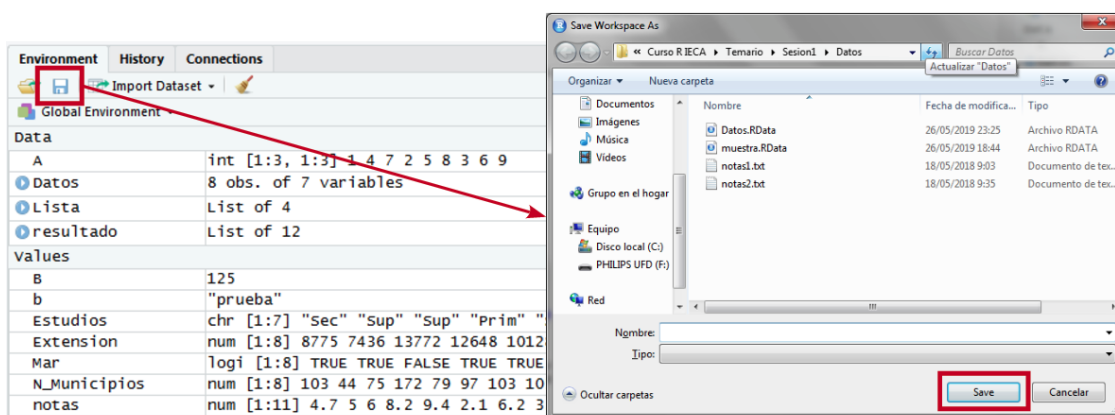
Además, todas estas acciones que hemos hecho para abrir un fichero, el programa las ha traducido a órdenes de R; en concreto ha creado la siguiente sentencia y la ha ejecutado automáticamente.

```
load("F:/Curso R IECA/Temario/Sesion1/Datos/Datos.RData")
```

que, como ya sabemos, sirve para cargar un archivo de tipo RData.

Para guardar los datos

Ya hemos visto como se carga un fichero, ahora vamos a ver **como se guardan los objetos** que tenemos creados en un fichero de tipo RData (recordad que éste es el tipo de fichero de R para los datos).



En la pestaña de **Environment** (Entorno) pulsamos sobre el icono del disco y navegamos hasta donde queremos que se guarde el archivo, le asignamos un nombre y pulsamos **Save**.

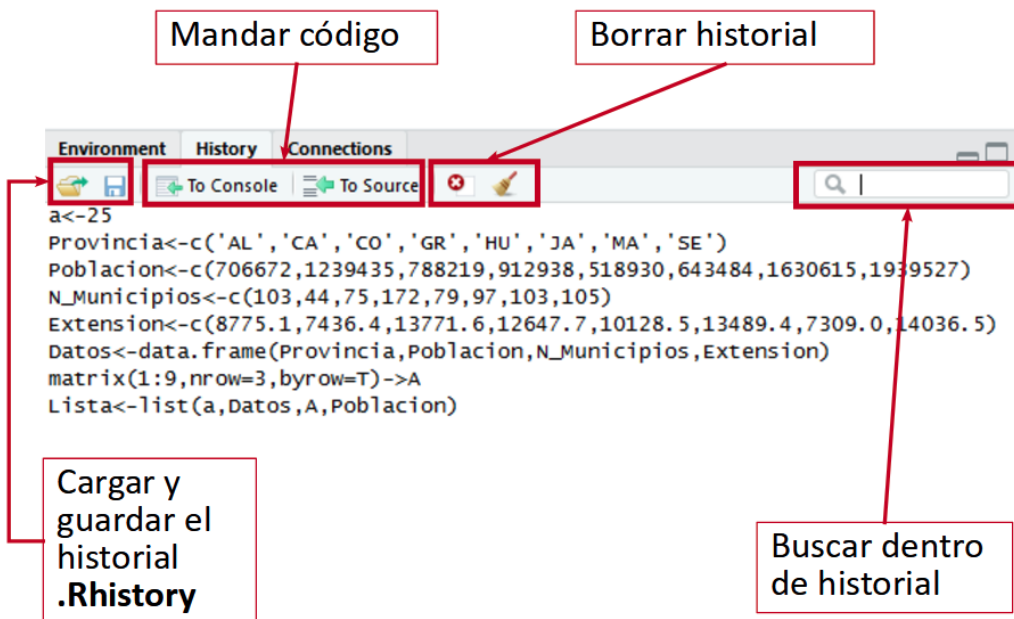
Esto salva los objetos en formato **.RData**.

2.3. Historial

En la pestaña **History** (Historial) podemos encontrar todas las órdenes que hemos ido ejecutando a lo largo de toda la sesión.

Las acciones que podemos hacer desde aquí son:

- Abrir un fichero de histórico con las órdenes, usando el icono de la carpeta
- Guardar el historial en un fichero (estos ficheros usan la extensión `.Rhistory`), usando el icono del disco.
- Volver a usar el código. Podemos usar el botón **To Console** para mandar la orden seleccionada a la consola o el botón **To Source** para mandarlo a un script si estamos haciendo un programa.
- Borrar el historial usando el icono de la escoba.
- Buscar alguna orden dentro de todo el historial.

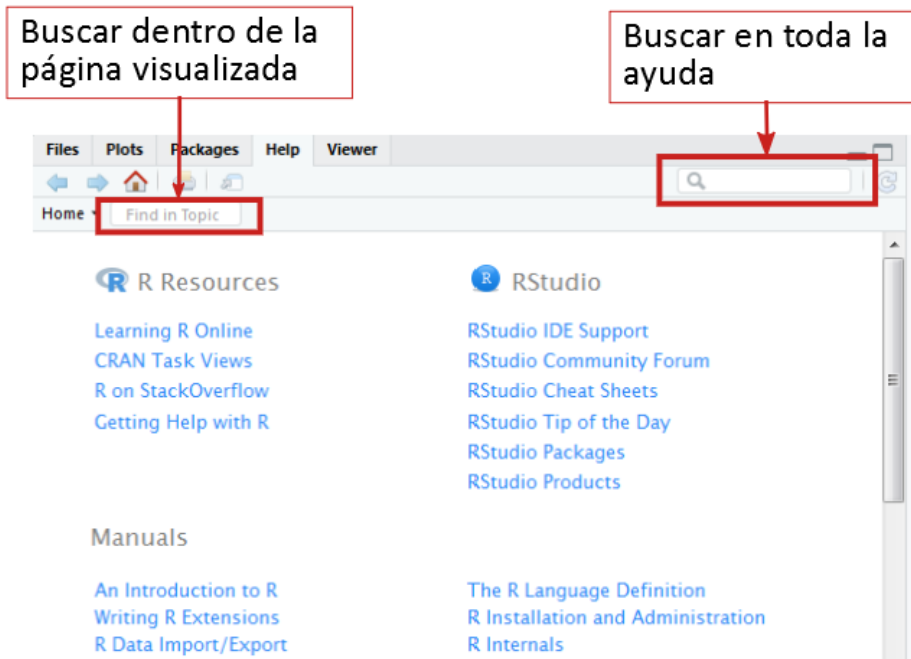


2.4. Ayuda

Otra de las pestañas que vamos a usar mucho cuando trabajemos con R es la pestaña de **ayuda**.

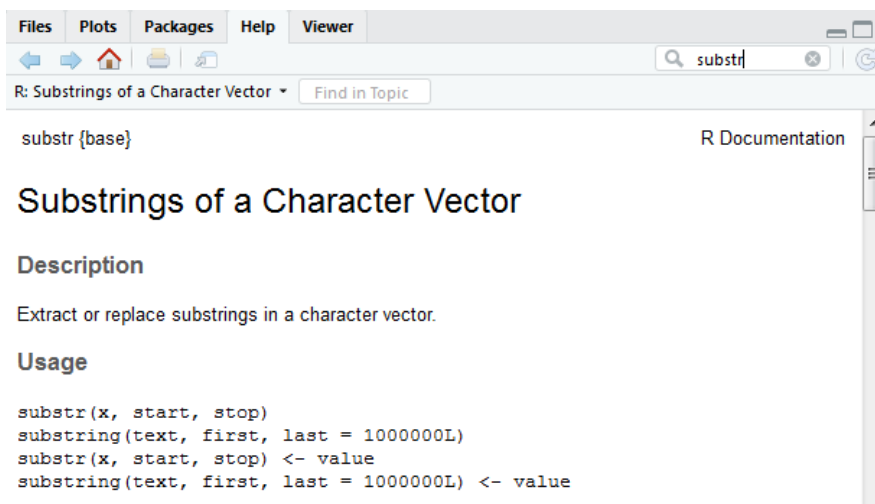
En ella podemos encontrar información sobre todas las funciones de R, así como ejemplos de uso de cada una de ellas.

Esta pestaña tiene dos buscadores que nos ayudarán a encontrar la información que necesitamos. Se puede **buscar dentro de toda la ayuda** o **dentro de la página que estamos consultado** en cada momento.



Un ejemplo de como se ve la ayuda de la función **substr()**.

Podemos ver una descripción de lo que hace la función, los parámetros que tiene, en que paquete se usa la función({base}) y ejemplos de como se puede usar.



Cuando buscamos en la ayuda una orden que pertenece a un paquete concreto, tenemos que tener el paquete cargado ya que si no, las órdenes no estarán disponibles en la ayuda.

Por ejemplo, si queremos ver la ayuda de la orden filter (que pertenece al paquete dplyr), si no tenemos cargado el paquete, no nos aparecerá en la ayuda la orden filter.

3. Gestión de paquetes

Los **paquetes** o librerías son pequeños programas independientes que están especializados en algún aspecto (realización de gráficos, tratamiento de datos, análisis estadístico, etc.) que podemos instalar en R para ampliar las capacidades del programa.

Para poder ser usados en R estos paquetes deben estar instalados y cargados en la memoria del programa.

Una de las ventajas que tiene R, frente a otros programas, es que cualquier persona puede crear un paquete y compartirlo con todo el mundo (actualmente cerca de 19.000 paquetes y creciendo).

Todos los paquetes disponibles se pueden encontrar en la web <https://cran.r-project.org/web/packages/index.html> para su descarga e instalación.

3.1. Paquetes interesantes

Como hemos dicho hay miles de paquetes. Aquí vamos a citar algunos de los más importantes en función de su uso. De algunos de ellos hablaremos en la unidad cinco del curso.

Para cargar datos

- **RODBC**. Acceder a Bases de Datos (Oracle, Access, Excel) y hacer consultas.
- **foreign**. Cargar datos en formatos de otro software como SPSS, SAS, DBF, etc.
- **readxl**. Cargar datos de Excel.
- **pxR**. Cargar datos de PC-Axis.
- **rjson**. Cargar datos en formato JSON.

Para procesar datos

- **dplyr**. Agregar, filtrar y aplicar funciones a datos.
- **lubridate**. Para trabajar con fechas.
- **reshape2**. Para transformar datos en una tabla.
- **stringr**. Para manipular texto.

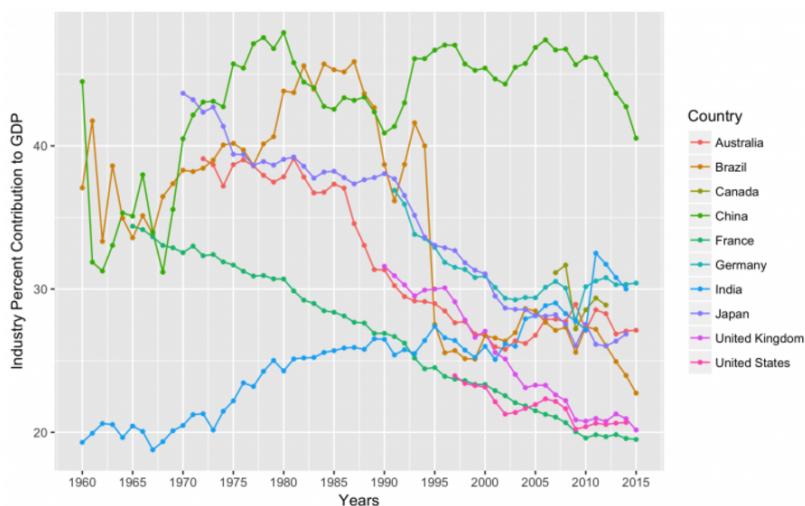
Para trabajar con datos espaciales

- **rgdal** y **sp**. Para tratamiento con datos espaciales.

Para la realización de gráficos

- **ggplot2**. Para mejorar las funciones gráficas básicas y la presentación de los gráficos.
- **lattice**. Para hacer gráficos en paneles.
- **RColorBrewer**. Paletas de colores predefinidas.

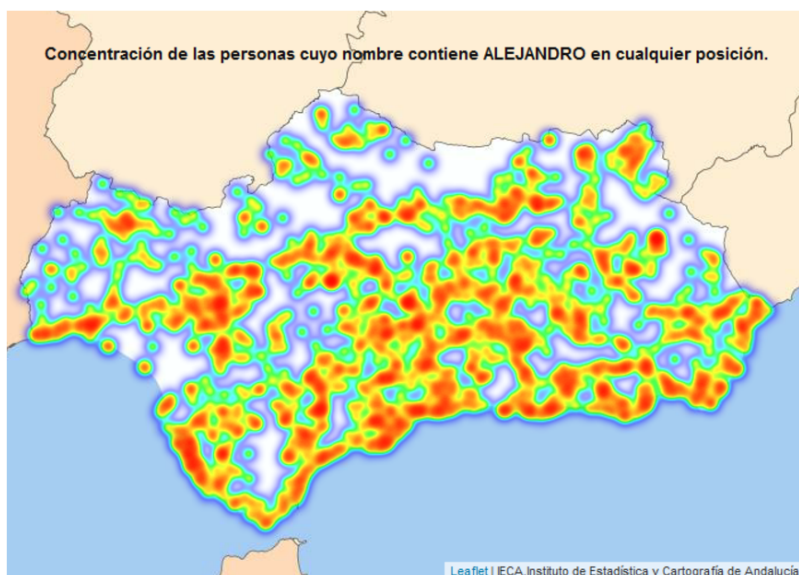
Gráfico realizado con ggplot2



Para presentar y difundir datos

- **shiny**. Aplicaciones interactivas para la presentación de resultados.
- **knitr**. Para realizar informes dinámicos en HTML.
- **rmarkdown**. Para realizar informes automáticos en distintos formatos.
- **leaflet**. Para la representación de datos espaciales en mapas interactivos.

Mapa realizado con leaflet

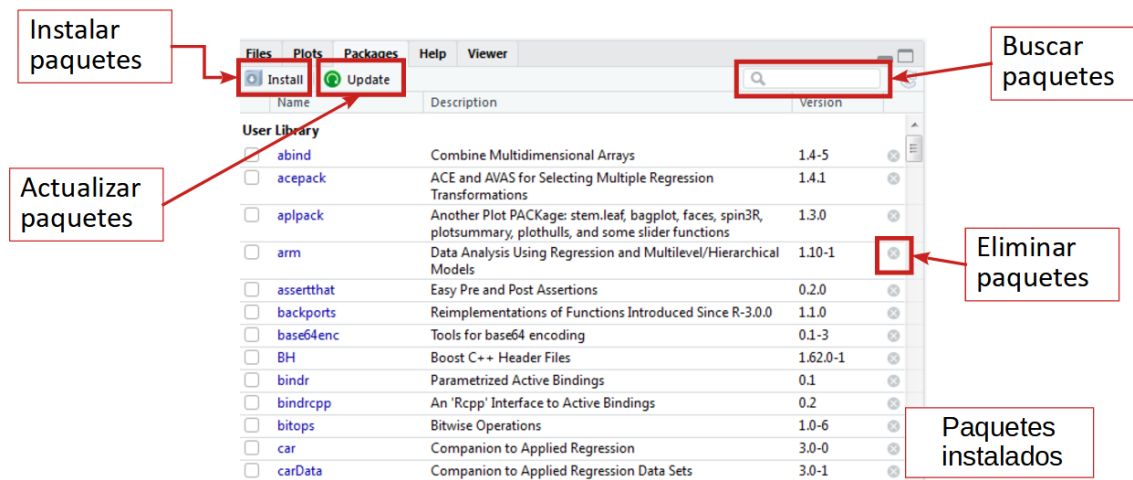


3.2. Operaciones sobre los paquetes

Para gestionar las librerías o paquetes podemos usar órdenes de R como:

- `install.packages()`. Instalar un paquete.
- `library()` o `require()`. Cargar un paquete.
- `detach()`. Descargar un paquete de la memoria.
- `remove.packages()`. Desinstalar un paquete.
- `old.packages()`. Ver los paquetes que están obsoletos.
- `update.packages()`. Actualizar paquetes.

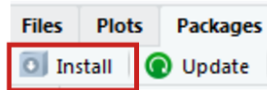
Pero RStudio nos permite realizar una gestión mucho más sencilla, mediante asistentes, desde la pestaña **Packages**.



3.3. Instalar paquetes

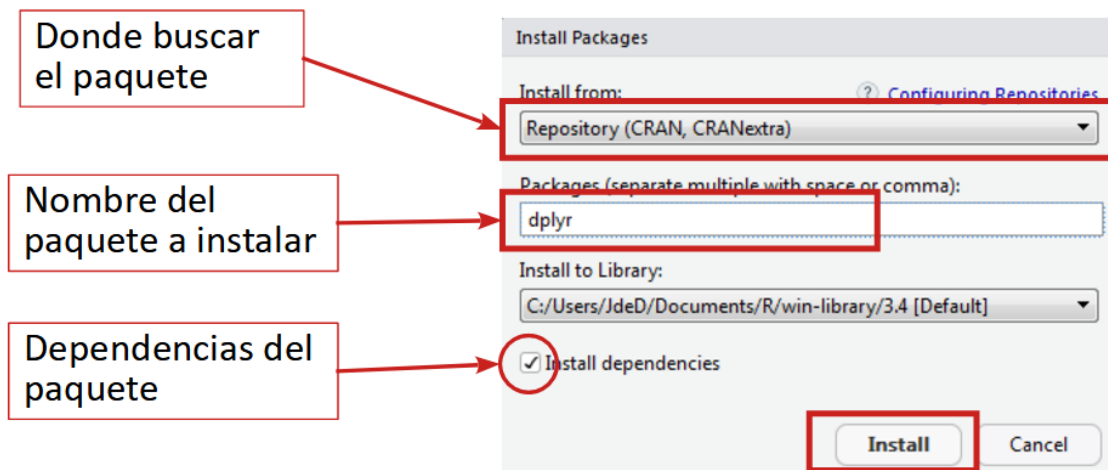
Para instalar un paquete se puede usar la orden **install.packages()** o bien usar el asistente.

Para hacerlo mediante el asistente de RStudio, tenemos que **pulsar el botón Install** de la **pestaña Packages**.

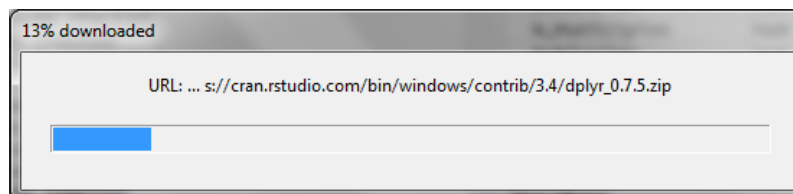


Una vez pulsado aparece una pantalla donde tenemos que definir una serie de parámetros.

- **Donde tiene que ir a buscar el paquete a instalar.** Por defecto busca en el Repositorio CRAN donde se encuentran todos los paquetes.
- **Como se llama el paquete a instalar.**
- **Donde se va a instalar.** La ubicación dentro del disco duro donde se guardará físicamente el paquete.
- **Install dependencies.** Esta marca es importante dejarla puesta siempre. Hay algunos paquetes que, para funcionar, necesitan que otros paquetes estén instalados (eso se llama dependencia). Si tenemos la marca activa, el programa buscará esos paquetes y los instalará, en el caso de que no estén instalados.



Una vez seleccionadas todas las opciones deseadas pulsamos el botón **Install** y comienza la descarga e instalación del paquete.

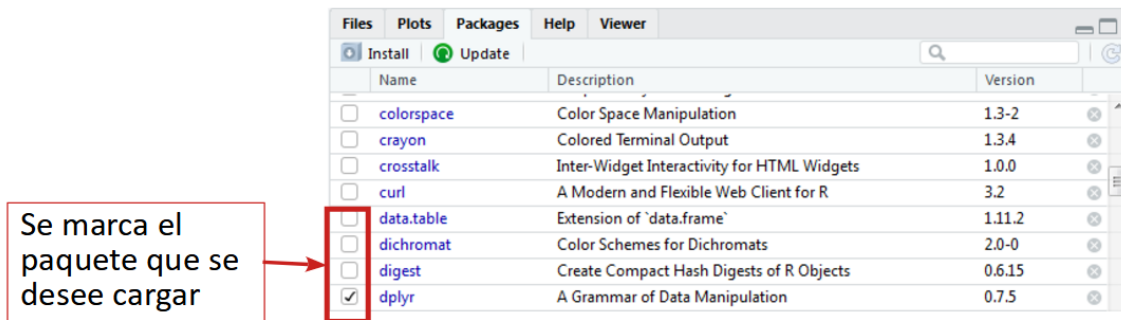


3.4. Cargar paquetes

Para poder usar todas las funcionalidades de un paquete en R antes debemos cargarlo en memoria. Hasta que no hagamos esto las funciones del paquete no estarán disponibles.

Podemos cargar un paquete con las órdenes `library()` o `require()` o bien usar el asistente de RStudio para hacerlo.

Desde el asistente simplemente tenemos que ir a la pestaña **Packages**, que es donde se encuentran todos los paquetes instalados en nuestro ordenador, y seleccionar el que queramos cargar.



Simplemente tenemos que marcar el paquete que deseamos cargar.

En caso de que ya no queramos usar el paquete, y lo queramos descargar de la memoria, solo tenemos que desmarcarlo. De esta forma liberamos espacio en la memoria para poder trabajar mejor.

Todos los paquetes que estén marcados son los que estarán cargados y disponibles para trabajar con ellos.

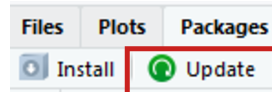
3.5. Actualizar paquetes

Como hemos dicho los paquetes son programas que debemos instalar y, con el tiempo, las personas que diseñaron los paquetes sacan nuevas versiones con mejoras.

Para poder contar con la versión más reciente de un paquete debemos mantenerlos actualizados. A la hora de actualizar hay que tener en cuenta que la versión del paquete que instalemos debe ser compatible tanto con la versión de R y como la de RStudio que tengamos instalada.

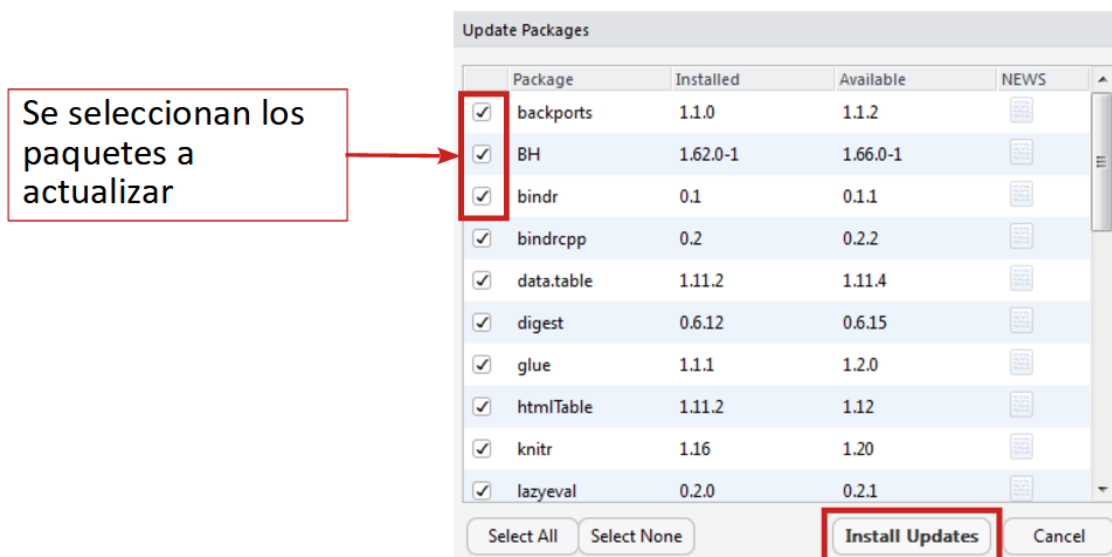
Para actualizar los paquetes podemos hacerlo mediante la orden **update.packages()** o podemos usar el asistente de RStudio.

Para hacerlo mediante el asistente de RStudio, tenemos que **pulsar el botón Update de la pestaña Packages**.

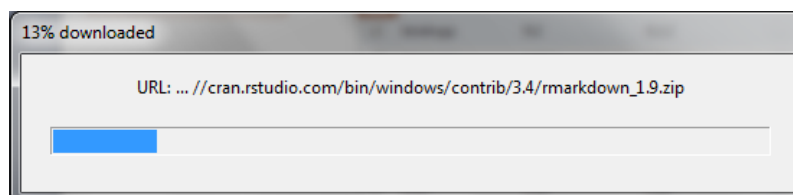


Una vez pulsado aparece una pantalla donde están todos los paquetes de los que se ha encontrado una actualización.

Simplemente tenemos que marcar los paquetes que queramos actualizar y pulsamos **Install Updates**.



Y comienza la descarga e instalación de la nueva versión del paquete.



4. Formas de trabajo

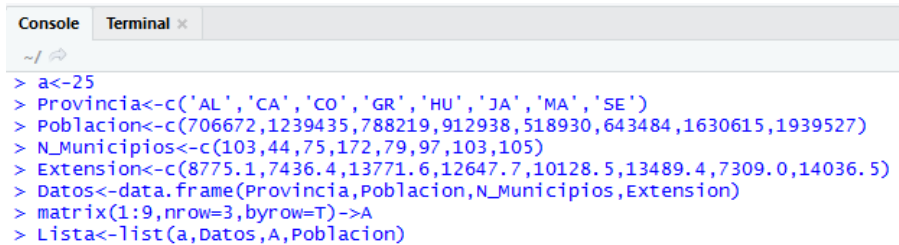
Cuando usamos R podemos trabajar de varias formas. Algunas de ellas ya las hemos visto con el programa R pero desde RStudio se añade una forma más que puede ser interesante para la gestión de nuestros proyectos.

En este apartado vamos a repasar las que ya conocíamos (**modo interactivo** y **script**) y a incluir el **trabajo por proyectos** propio de RStudio.

4.1. Modo interactivo

El modo interactivo ya lo conocemos de R. Se trata de escribir directamente las órdenes en la consola y ejecutándolas directamente pulsando ENTER o INTRO en nuestro ordenador.

En este caso solo trabajamos sobre la consola de RStudio.



The screenshot shows the RStudio interface with the 'Console' tab selected. The code entered in the console is as follows:

```
> a<-25
> Provincia<-c('AL','CA','CO','GR','HU','JA','MA','SE')
> Poblacion<-c(706672,1239435,788219,912938,518930,643484,1630615,1939527)
> N_Municipios<-c(103,44,75,172,79,97,103,105)
> Extension<-c(8775.1,7436.4,13771.6,12647.7,10128.5,13489.4,7309.0,14036.5)
> Datos<-data.frame(Provincia,Poblacion,N_Municipios,Extension)
> matrix(1:9,nrow=3,byrow=T)->A
> Lista<-list(a,Datos,A,Poblacion)
```

4.2. Modo Script

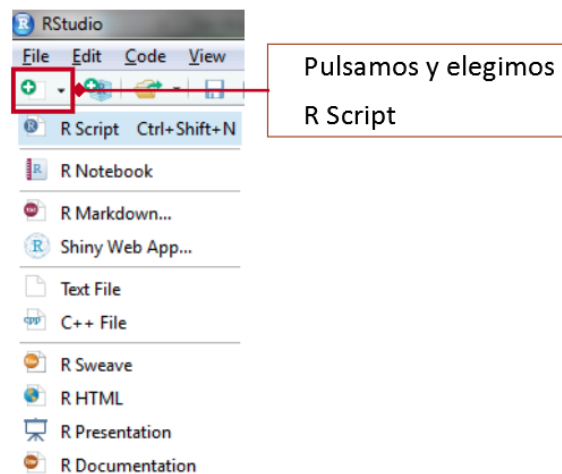
Esta forma de trabajar también la conocemos de R. Se trataría de abrir un documento donde vamos escribiendo las órdenes necesarias para completar una tarea.

Ese conjunto de órdenes no se ejecuta hasta que lanzamos el programa al completo. Lo que cambia respecto a R es la forma de acceder a los scripts desde RStudio.

Para trabajar mediante scripts con RStudio hay que seguir una serie de pasos.

1. Abrir documento para script (R Script)

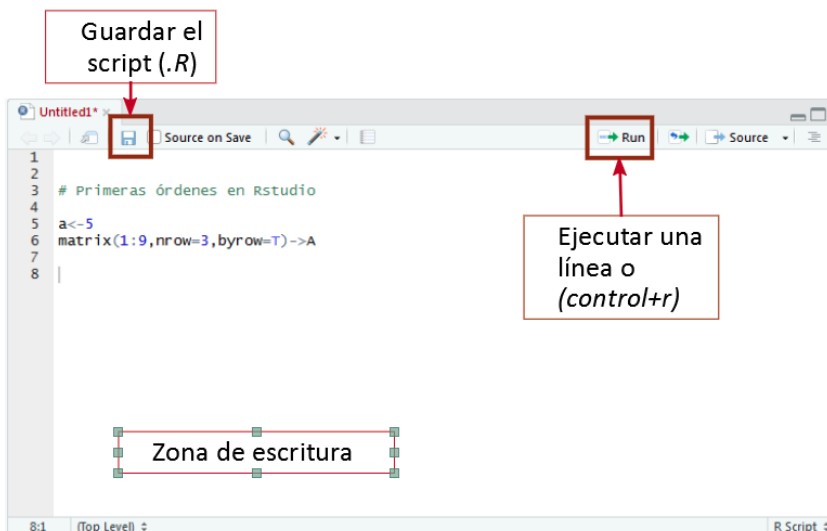
En RStudio podemos acceder desde **File** → **New File** → **R Script** o pulsando el icono desde la barra de herramientas.



2. Escribir el conjunto de órdenes

Una vez abierto el documento podemos empezar a escribir las órdenes pero de una forma más sencilla que en el propio programa R.

RStudio nos ofrece una serie de ayudas como indentado de funciones, resaltado de parámetros, autocompletado de funciones, numeración de líneas, etc. que nos van a facilitar el trabajo con R.



Una vez que hemos terminado el programa o script podemos ejecutarlo pulsando el botón **RUN** o las teclas **control+r**.

De esta forma se va ejecutando la línea sobre la que estemos en ese momento. Si queremos ejecutar varias líneas de código al mismo tiempo simplemente tenemos que seleccionar las líneas a ejecutar y pulsar RUN.

4.3. Proyecto de R

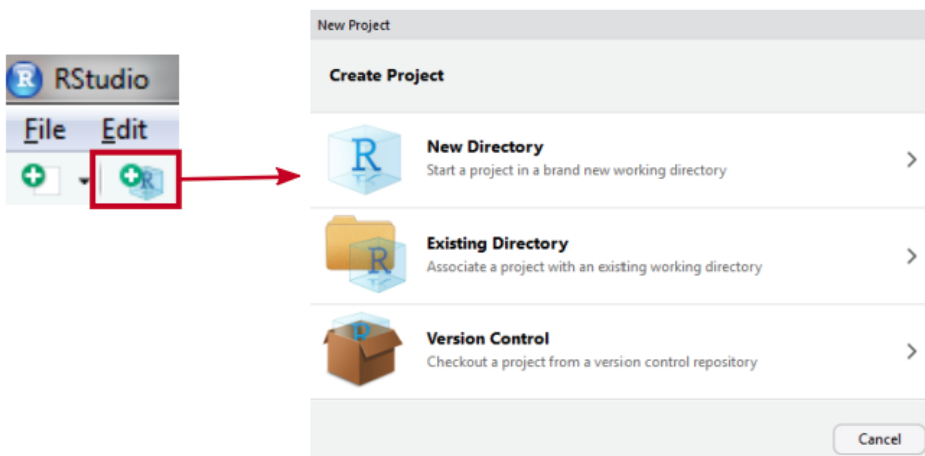
Un **proyecto** es una nueva forma de trabajo que nos ofrece RStudio. Es una filosofía más ordenada que hace que sea más fácil dividir el trabajo en múltiples contextos: cada uno con su propio directorio de trabajo, su espacio de trabajo, su historial y todos los ficheros de origen que usemos.

Antes de empezar a trabajar en un proyecto se lo tenemos que indicar al programa para que éste prepare el contexto de trabajo creando el directorio de trabajo, las carpetas donde se van a almacenar los archivos y varias opciones más que tenemos que indicar al principio.

Para trabajar como un proyecto tenemos que seguir los siguientes pasos:

1. Crear el proyecto

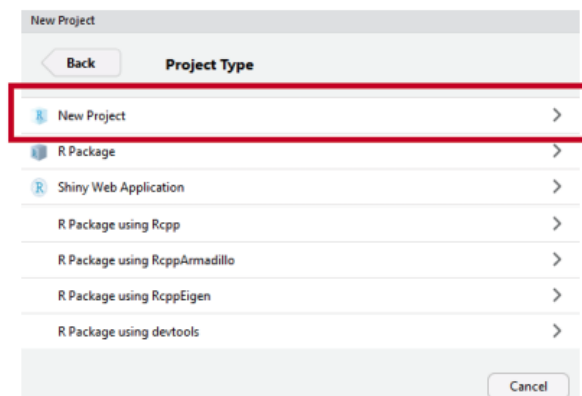
Para ello podemos acceder desde el menú **File** → **New Project** o pulsado el icono correspondiente en la barra de herramientas.



Después elegimos si vamos a crear un nuevo directorio (opción recomendable si empezamos un nuevo proyecto) o vamos a usar alguno que ya exista.

2. Elección de tipo de proyecto

Después tenemos que elegir el tipo de proyecto que vamos a realizar. En el punto que estamos nuestra opción sería **New Project** ya que el resto de opciones son para usuarios avanzados.



3. Asignamos nombre y ubicación

Por último, le asignamos un nombre al proyecto y elegimos la ubicación en la que se va a almacenar todo lo que se vaya generando en la sesión de trabajo.

Back

Create New Project



Directory name:

Nombre del proyecto

Create project as subdirectory of:

~/Copia

Browse...

☐ Open in new session

Create Project

Cancel