

Primeros pasos con R



R es un lenguaje de programación creado a principios de los años 90, enfocado al análisis estadístico de datos. Nació como una ampliación de un lenguaje previo, llamado S, y poco a poco se ha ido convirtiendo en uno de los lenguajes más utilizados en el análisis científico e incluso aprendizaje automático.

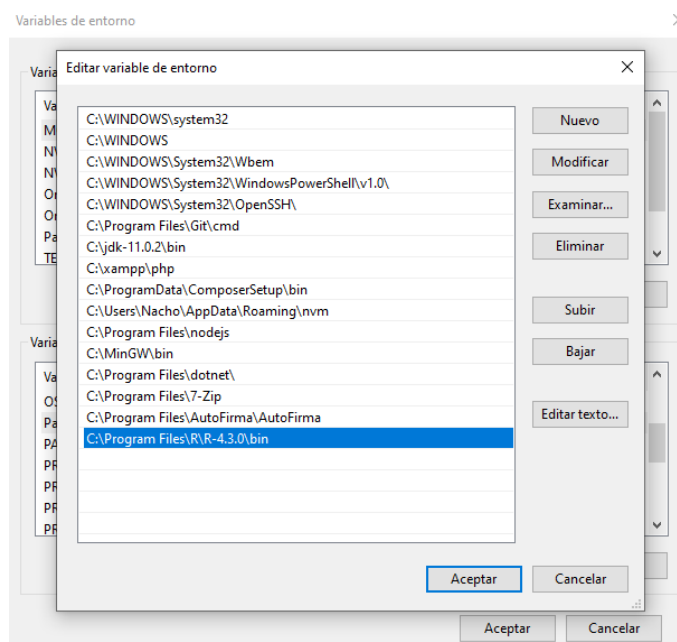
Algunas de sus características más destacadas son:

- Dispone de un amplio abanico de módulos adicionales para cálculos estadísticos
- Permite orientación a objetos (aunque no es obligatoria)
- Es multiplataforma
- Puede integrarse con varias bases de datos

Podemos consultar más información acerca del lenguaje en su [web oficial](#).

1. Instalación de R

Para instalar R en el sistema, debemos ir a [esta web](#), y descargar la versión correspondiente a nuestro sistema operativo desde el *mirror* que nos resulte más cómodo. Convendrá tener el intérprete (comandos `r` y/o `rscrip`) añadido al PATH del sistema. En el caso de Windows, podemos añadir la carpeta *bin* de la instalación de R a la variable de entorno *Path*.



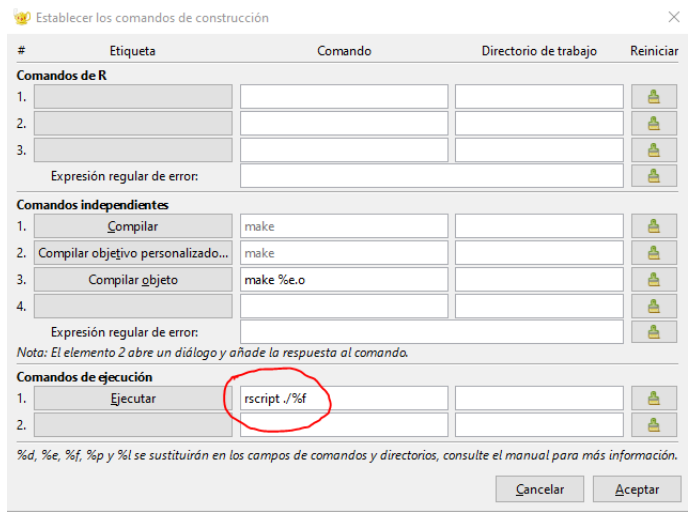
2. Elegir un IDE apropiado

Existen varias alternativas para poder desarrollar programas en R. Veamos algunas de las opciones disponibles.

2.1. Utilizando Geany

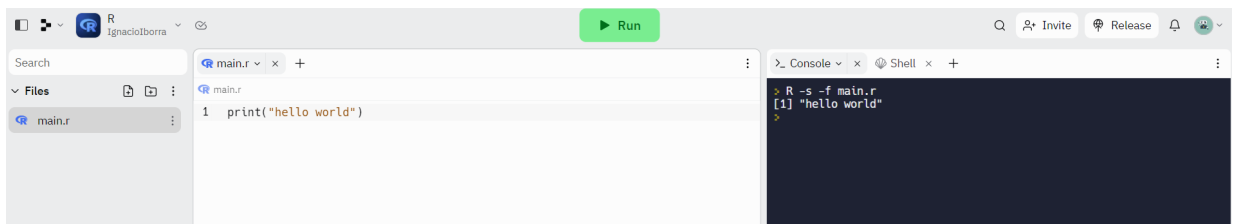
Si ya tenemos descargado e instalado R, el intérprete del lenguaje estará ya correctamente configurado en nuestro sistema. En este caso, podemos emplear un IDE de propósito general como [Geany](#) para editar nuestros programas. Podemos instalarlo fácilmente desde su web oficial en Windows y en Mac. En el caso de Linux es mejor instalar el paquete *geany* junto con sus dependencias desde el gestor de dependencias del sistema operativo.

La mecánica es sencilla: crearemos archivos nuevos con la extensión apropiada (.r), editamos el código y luego lo ejecutamos. Debemos configurar Geany para que ejecute correctamente el programa. Una vez tengamos nuestro primer archivo R creado y guardado, vamos al menú *Construir > Establecer comandos de construcción* y establecemos el comando de ejecución de este modo:



2.2. Intérpretes online

Una opción alternativa para desarrollar programas en R, especialmente cuando son relativamente simples y cortos, es utilizar algún editor online, como [éste](#). Es posible que nos pida un registro (gratuito) y tendremos un espacio de trabajo online para crear y probar nuestros programas en R. Simplemente debemos escribir el código en el panel correspondiente, y hacer clic en el botón para ejecutar la aplicación, viendo el resultado en la consola derecha.



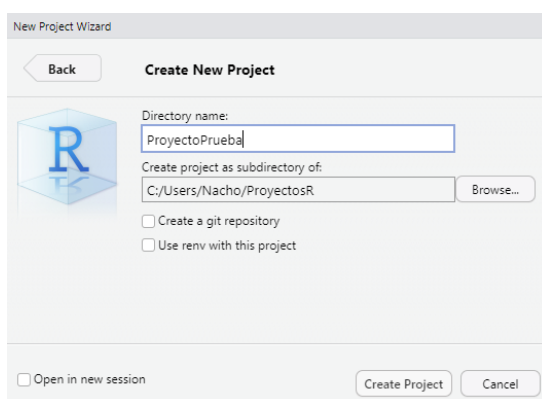
Alternativamente, existen otros editores online que no requieren de registro, como [este de W3Schools](#).

2.3. RStudio

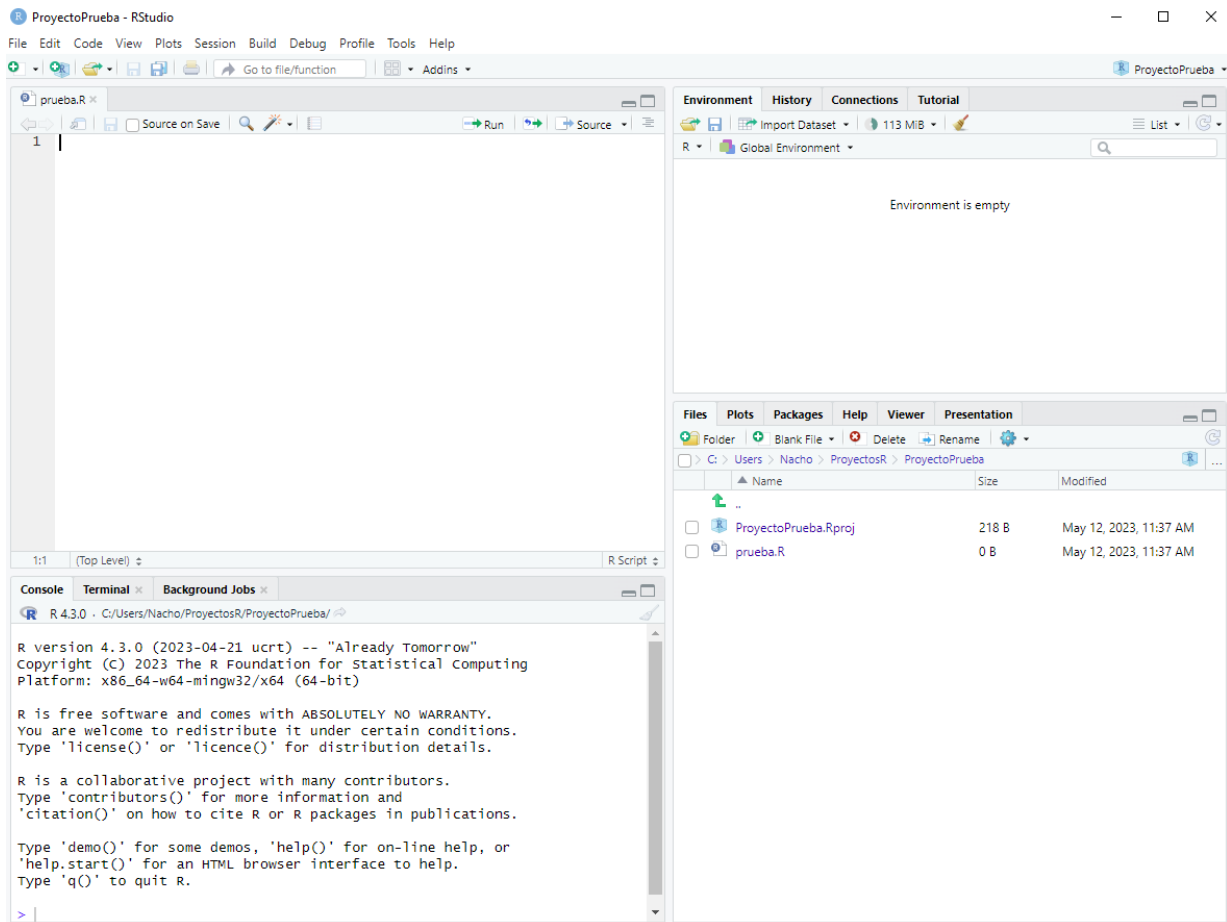
Pasamos a ver el editor más utilizado para desarrollo de aplicaciones en R, llamado **RStudio**. Podemos descargarlo e instalarlo desde su [web oficial](#). En el caso de Mac, es posible que también necesitemos instalar el paquete [XQuartz](#) para poder abrir este IDE.

Cuando iniciamos el IDE por primera vez, nos preguntará qué versión de R queremos utilizar (de hecho, podemos decargar *RStudio* junto con R). Podemos elegir la versión que ya hemos instalado previamente en nuestro sistema. Una vez iniciado, para crear programas vamos al menú *File > New File > R Script*. Podemos elegir dónde guardar el fichero con *File > Save As*, y se guardará con la extensión apropiada.

Alternativamente, podemos crear **proyectos**, con *File > New Project*. Nos dejará elegir un nombre de proyecto, y en qué carpeta lo queremos ubicar.



Todos los archivos que creamos se definirán dentro de la carpeta del proyecto, y podremos verlos en el panel inferior derecho.



En la sección de *Console* podemos probar instrucciones simples y ver el resultado, y en la sección *Environment* veremos las variables que tengamos creadas en cada momento, con sus valores.

Para ejecutar nuestros programas (una vez abiertos en RStudio), seleccionamos todo el código (o la línea que queramos ejecutar) y pulsamos el botón *Run* de la parte superior del editor, o pulsamos *Control + Intro*.

2.4. Google Colab

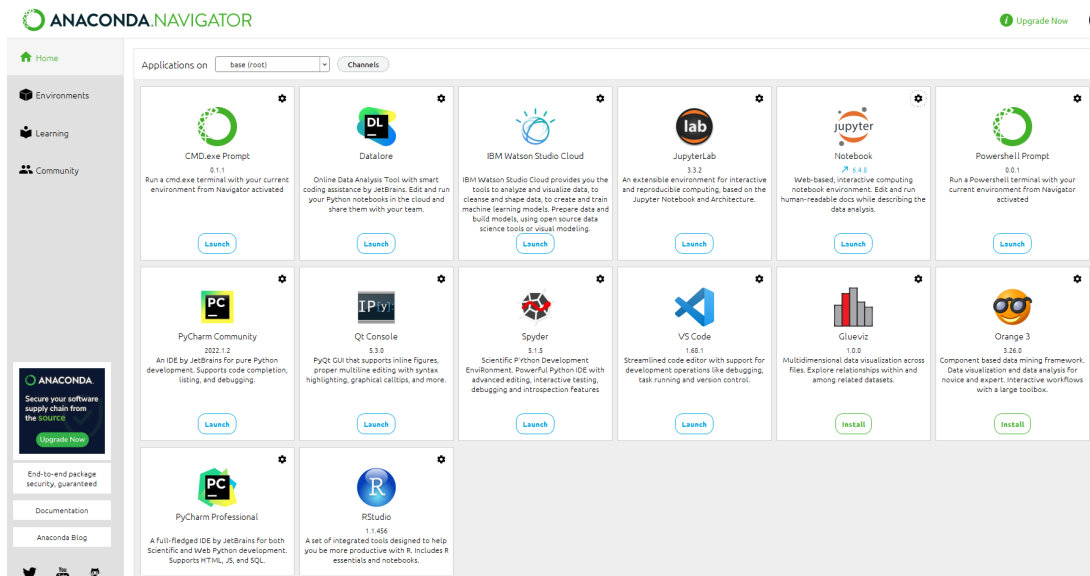
Google Colab es una herramienta ofrecida por la nube de Google que permite definir documentos (*notebooks*) que intercalan código y texto. Esto permite, entre otras cosas, definir tutoriales guiados y auto-explicativos sobre ciertos conceptos. Aunque fundamentalmente está pensado para Python, también admite trabajar en algún otro lenguaje, como es el caso de R, a través de [este enlace](#).

En [este documento](#) está explicado el funcionamiento general de Google Colab, en el caso de Python. Para R el caso es similar, y se incorporan algunas librerías habituales del lenguaje, como *dplyr* o *tidyverse*. Podemos instalar paquetes adicionales del mismo modo que lo haríamos en local, como se explica en [este documento](#).

2.5. Anaconda

Anaconda es un pack de software que integra diferentes herramientas orientadas al trabajo con lenguajes como R o Python, en ámbitos como *data science* o *machine learning*. Entre otras cosas, podemos instalar el propio entorno R/Python, junto con algunos editores predefinidos (como RStudio o PyCharm) y también

algunas librerías típicas de *data science* o *machine learning* tales como *NumPy* o *TensorFlow*. Podemos obtener más información [aquí](#).



NOTA: Anaconda incorpora su propia versión de Python y/o lenguaje R, que no interfiere con la que podamos tener instalada en el sistema previamente.

3. Nuestro primer programa R

Para terminar esta introducción al lenguaje, veamos cómo se escribiría el típico primer programa que muestre un mensaje de *Hola mundo* por pantalla. Crearemos para ello un archivo fuente, por ejemplo, `prueba.r`, con el siguiente contenido:

```
print("Hola mundo")
```

Para ejecutarlo desde nuestro IDE:

- Si usamos Geany, pulsamos el botón de *Ejecutar* de la barra de herramientas
- Si usamos el editor online de *repl.it*, pulsamos el botón de ejecutar o *Run*
- Si usamos RStudio, seleccionamos la(s) línea(s) a ejecutar y pulsamos *Run* o *Control + Intro*

El resultado en cualquier caso será el mismo, viendo por terminal el texto que hemos indicado en la instrucción `print`:

```
Hola mundo
```