

Fundamentos para el uso de R/RStudio

Alejandro Franco y Luis Carlos Ramos

28/Junio - 23/Julio

Bienvenid@s

Este documento aborda los temas que serán tratados durante el curso con título *Fundamentos para el uso de R/Rstudio*, de manera que el asistente tenga acceso facilitado a la información esencial. Los detalles de cada tema serán desarrollados durante las sesiones, así como las dudas que surjan de los ejercicios que aquí se proponen.

Especificaciones

1. Las letras en **negrita** y *crusiva* contienen links que podrían ser de interés para su visita, como *este*, que contiene archivos de guías rápidas para las aplicaciones más comunes de R y los programas con los que se le asocia.
2. **IMPORTANTE:** así es como se mostrará una indicación a seguir, no la pierdas de vista.
3. Un renglón con sangría indica el inicio de un ejercicio extraclase que se propone para reforzar el tema.

R/RStudio

R es un lenguaje de programación que tiene como precedente al lenguaje S y actualmente es usado para aprender estadística, enseñarla, hacer proyecciones en mapas, trabajar con imágenes, procesar datos genómicos, manipulando datos, obtener gráficos, estadísticos y simulaciones en una misma aplicación. Se distribuye como un software gratuito y de código abierto, compatible con los sistemas operativos Windows, macOS y Linux. El software para el uso del lenguaje se descarga directamente del CRAN (*Comprehensive R Archive Network*) en *esta página web*, que mostrará la siguiente ventana, donde debe descargarse el ejecutable de acuerdo al sistema operativo y características del equipo.

Al terminar de ejecutar la instalación y abrir el programa, se mostrará la interfaz gráfica (GUI, *graphical user interface*) de R (Fig. 2), que puede resultar un poco abrumadora si es la primera vez que utilizamos un software de programación. Para hacer más dinámico el manejo del lenguaje, se hace uso de un entorno de programación, o IDE (Integrated Development Environment). En R las tres plataformas gráficas principales utilizadas en el manejo de R son **RStudio**, **RCommander** y **RKward**, sin embargo existen más. El que se utiliza con más frecuencia es RStudio y es el que será usado durante el curso.

IMPORTANTE: El orden de ejecución de los programas es importante, primero debe ejecutarse R y después RStudio, para un funcionamiento adecuado.



Cran
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

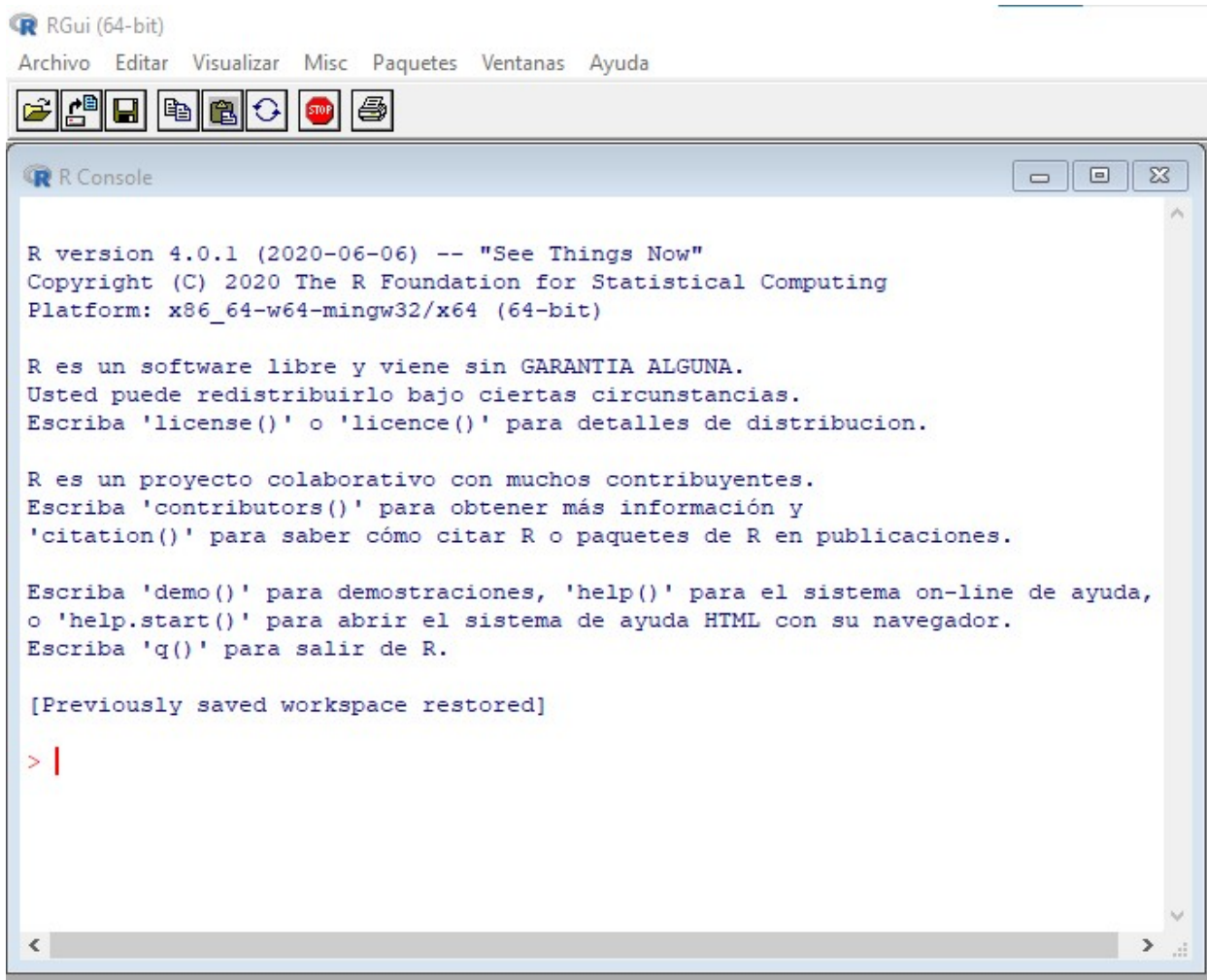
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2021-05-18, Camp Pontanezen) [R-4.1.0.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Figure 1: Ventana de la página web que muestra los ejecutables por sistema operativo



Es necesario entender como se compone el espacio de trabajo en RStudio para hacer uso eficiente de las facilidades que provee, la consola, ambiente, editor de código, las utilidades y el menú superior son las

partes principales en que se divide RStudio (Fig. 3).

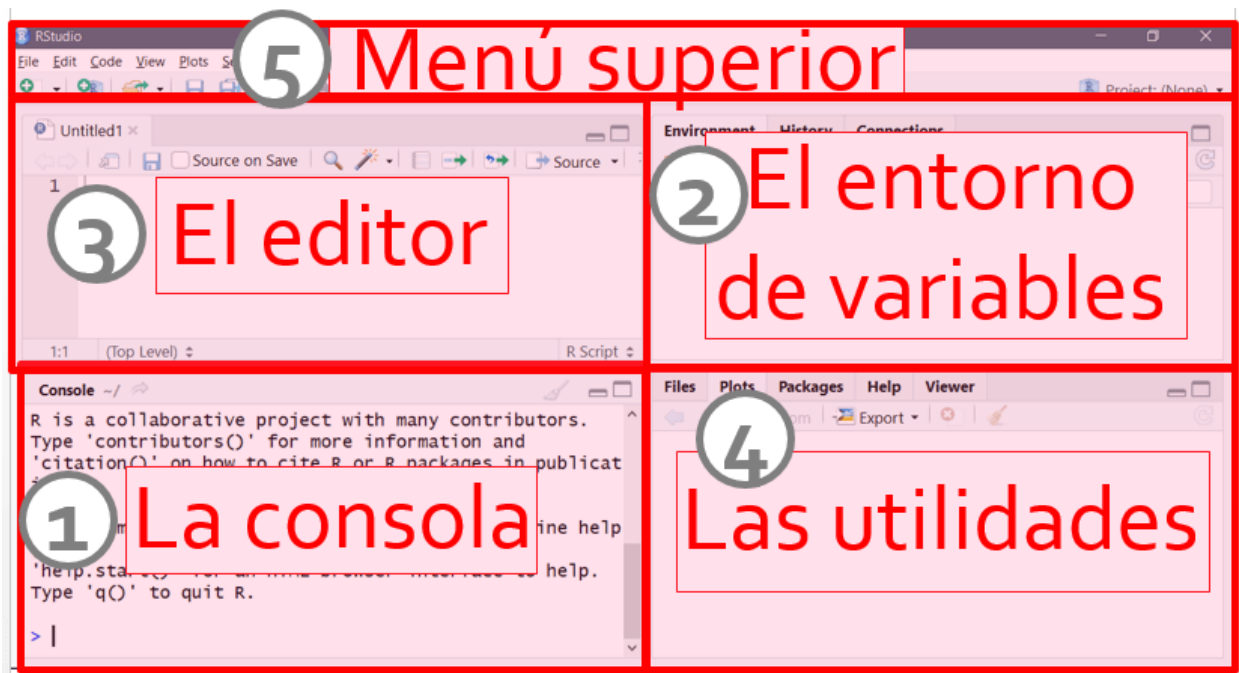


Figure 2: Divisiones principales de la ventana de RStudio

Control de versiones

Trabajar a través de un *sistema de control de versiones* facilita el intercambio y resguardo de información en una plataforma virtual, como los conocidos Microsoft OneDrive y Google drive, que al mismo tiempo que almacenan tus archivos, te permiten trabajar de manera remota y colaborativa en tiempo real. Para cumplir esa función en R, se usa **GitHub**, accede al enlace y crea una cuenta en la plataforma, asegurandote de registrar un correo que sí ocupes, pues será requerido más adelante. Antes de poder usar GitHub como un control de versiones es importante familiarizarse con los siguientes términos:

- **Repository:** equivalente a un 'folder/directorio'. Todos los archivos
- **Commit:** significa guardar las ediciones y cambios hechos. Git compara la versión anterior de todos los archivos en el 'repo'. Son 'el corazón' de los sistemas de control pues se observan todos los cambios que se han hecho con el tiempo
- **Push:** actualizar el repositorio con las ediciones hechas, así es posible tenerlos en línea
- **Pull:** actualizar la versión local del archivo para tener la última versión que alguien haya modificado (trabajo colaborativo)
- **Staging:** preparar un archivo para 'commit'. Cuando se tienen múltiples archivos modificados de un repositorio, es más eficiente 'stage' cada uno por separado para 'commit' con el comentario de cambio respectivo
- **Branch:** cuando un mismo archivo tiene dos copias de forma simultánea. Se crea una 'branch' cuando 'pull' el documento para su edición local

- **Merge:** ediciones independientes del mismo archivo que son incorporadas a uno solo y unificado archivo.
- **Conflict:** más de una persona realizó un cambio en una misma ‘sentencia’, eso no puede ser ‘merge’ automáticamente, de tal forma que se hace manualmente
- **Clone:** hacer una copia de un repositorio existente y editarlo personalmente. Es cuando te integras a un equipo de trabajo
- **Fork:** copia personal de un repositorio ya existente.

Realiza el ejercicio que se propone ***aquí*** para familiarizarte más rápido con los términos.

Para usar GitHub como control de versiones no basta con tener una cuenta, debe crearse un enlace entre la plataforma y el IDE de R *RStudio*. Para ello, es necesario descargar ***GIT en este enlace__***.

Tipos de objetos

Caracteres

Números complejos

Enteros

Vectores

Lógicos

Matrices

Listas

Data frame y arrays

Factores

Funciones

Valores especiales

Faltantes

Infinito

Tiempo (Fechas y horas)

Operaciones

Aritméticas

Lógicas

Relacionales

Paquetes

Subconjuntos de datos

Extracción de datos

dplyr

Sistemas de graficado

Graficado base

ggplot2

Pruebas estadísticas