

Club de lectura

"Efficient R programming", de Colin Gillespie y Robin Lovelace.

CONFIGURACIÓN EFICIENTE

Capítulo 2

Fecha: Martes, 19 de abril

Hora: 18:00 a 19:00 UTC-5

Lugar: **zoom**

Inscripción: *meetup*



Conferencistas:



Sara Garcés



Alejandra
Londoño

Organiza:



Semillero
Unal

Colaboran:



R-Ladies Medellín +
Barranquilla + Galápagos

"Efficient R programming", de Colin Gillespie y Robin Lovelace.

Capítulo 2
**CONFIGURACIÓN
EFICIENTE**

Agenda:

1.R y los sistemas operativos

Monitoreo del sistema en Linux, Mac y Windows

2.Versión R

Cómo mantener actualizada la instalación y los paquetes de R base

3.Puesta en marcha de R

4.Consejos y Recomendaciones

R y los sistemas operativos


Sistema Operativo:

R es compatible con los tres tipos principales de sistemas operativos (OS): Linux, Mac y Windows.

La información básica específica del sistema operativo se puede consultar desde R utilizando **Sys.info()**



*Paquete Necesario:
library("benchmarkme")*



Uno de los problemas relacionados con el sistema operativo a considerar, son dependencias externas: programas de los que dependen los paquetes de R. A veces, las dependencias de paquetes externos deben instalarse manualmente (es decir, **no usar** **install.pack ages()**).


R y los sistemas operativos

Supervisión de sistemas operativos y recursos

La supervisión del sistema es un tema complejo que se extiende a la administración del sistema y la administración del servidor.



En Linux: La parte superior del comando shell muestra las cifras clave de uso de recursos para la mayoría de las distribuciones.

En Mac: El Monitor de actividad proporciona una funcionalidad similar. Esto se puede iniciar desde la carpeta Utilidades en Launchpad. 



En Windows: El Administrador de tareas proporciona información clave sobre el uso de RAM y CPU por proceso. Esto se puede iniciar en las versiones modernas de Windows presionando Ctrl- Alt-Supr o haciendo clic en la barra de tareas e Iniciar el Administrador de tareas.

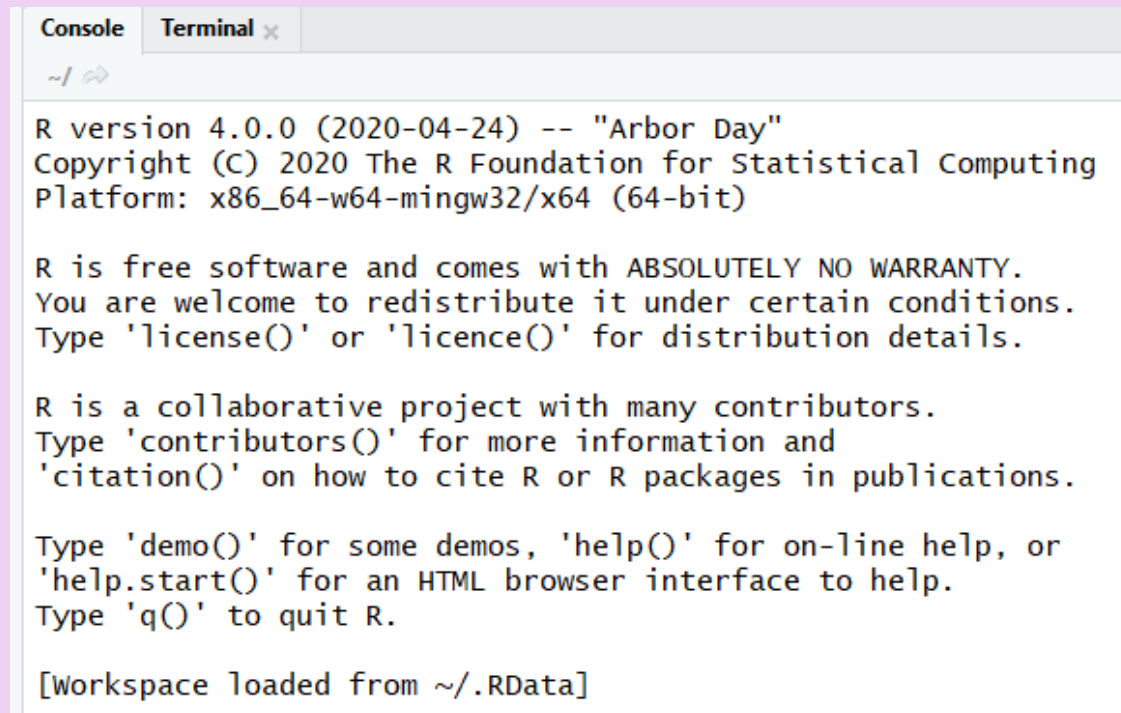
Versión de R

¿Cómo cambiar la versión de R en RStudio?

1

Abrir RStudio

Al abrir RStudio, vamos a ver en el panel inferior, lo que se denomina consola, podremos comprobar la versión con la que estamos trabajando.



```
Console Terminal x
~/
R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]
```

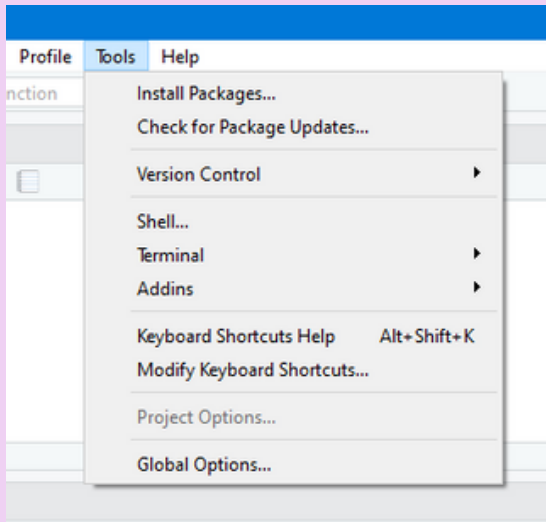
Versión de R

¿Cómo cambiar la versión de R en RStudio?

2

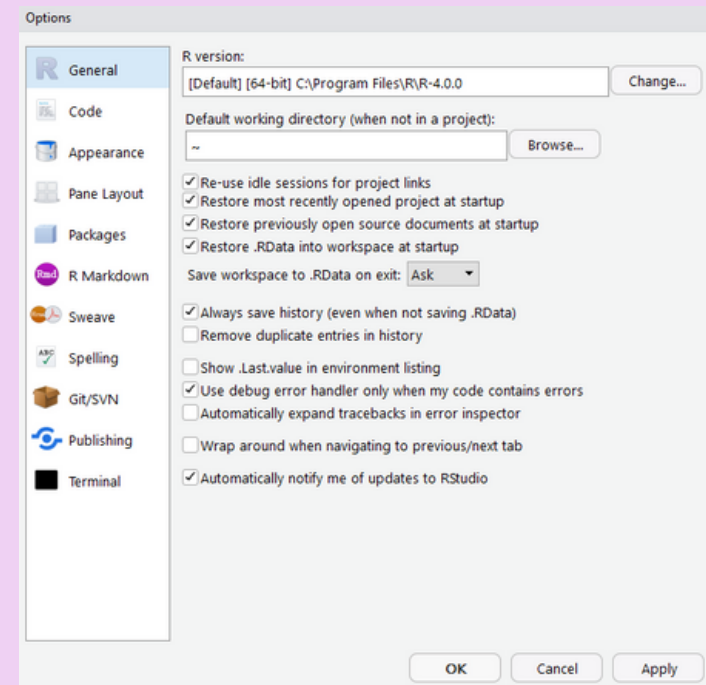
Opción Tools:

Vamos a la pestaña de Tools dentro de esta ventana, accedemos a Global Options.



3

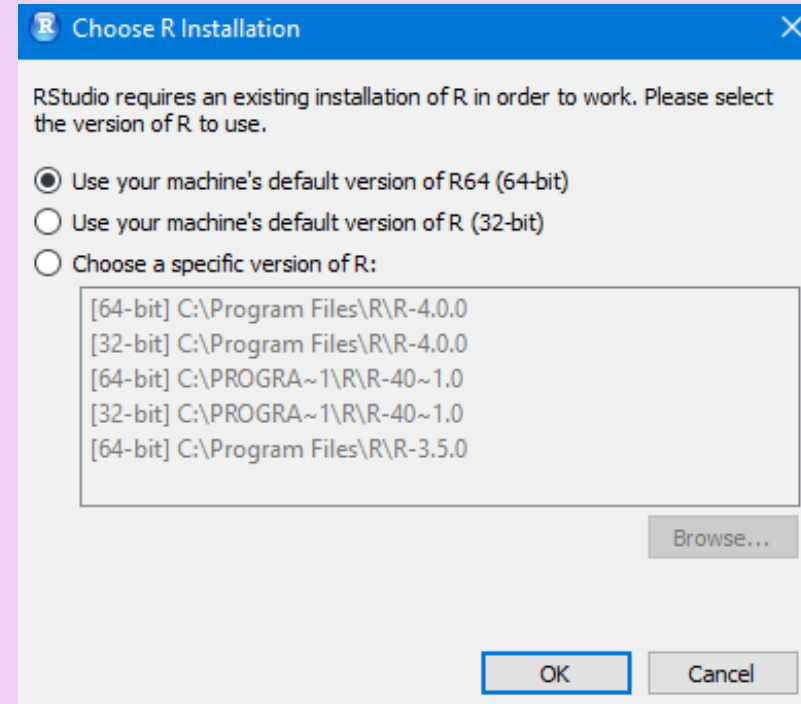
Selección de la versión de R en RStudio:



Versión de R

¿Cómo cambiar la versión de R en RStudio?

- 4 Cambiar versión de R en RStudio:** Al darle a la opción de cambiar la versión de R dentro de RStudio, se nos abre una nueva ventana donde veremos todas las versiones que tenemos instaladas y donde podremos seleccionar la versión de R con la que queremos trabajar.
Para ello, seleccionaremos la opción de Choose a specific version of R.



- 5 Reiniciar el Programa**

Versión de R

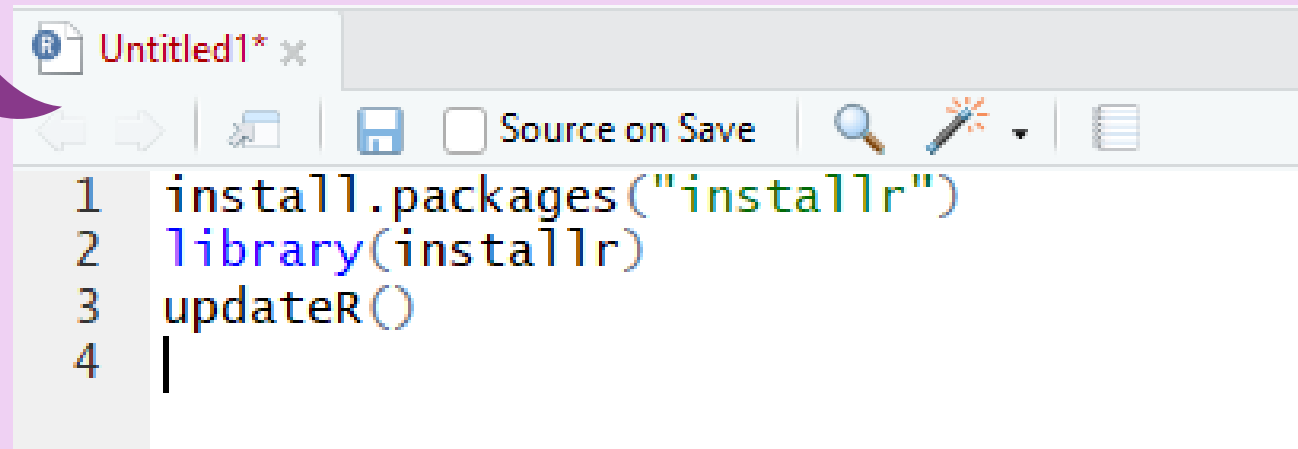
Instalación de R

El método de instalación de R varía para Windows, Linux y Mac.

En **Windows**, un único archivo **.exe** alojado en **<https://cran.r-project.org/bin/windows/base/>** , el cual instalará el paquete R base.

Solo para el sistema operativo Windows.

Actualización de R

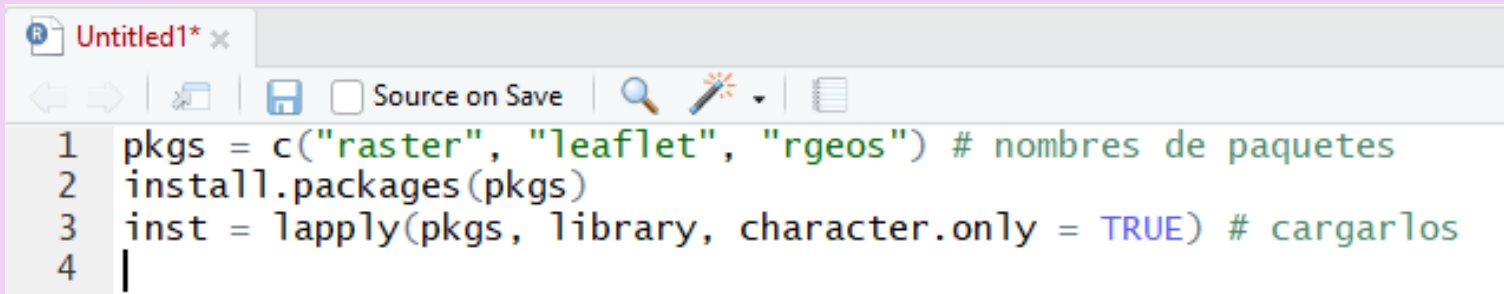


```
1 install.packages("installr")
2 library(installr)
3 updateR()
4 |
```


Versión de R

Instalación de paquetes de R

Los proyectos grandes pueden necesitar varios paquetes para ser instalados. En este caso, los paquetes requeridos se pueden instalar a la vez. Esto se puede hacer de manera rápida y concisa con el siguiente código:



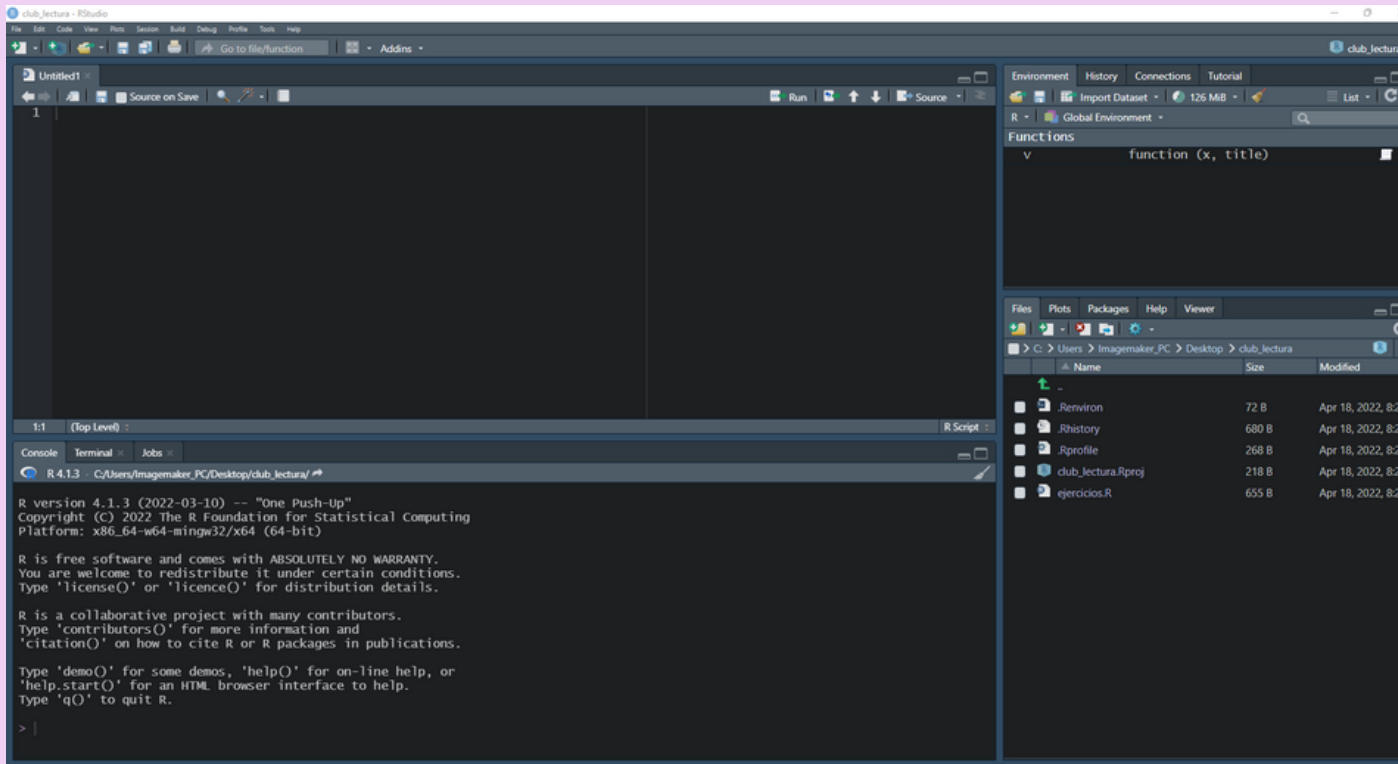
```
1 pkgs = c("raster", "leaflet", "rgeos") # nombres de paquetes
2 install.packages(pkgs)
3 inst = lapply(pkgs, library, character.only = TRUE) # cargarlos
4 |
```

Actualización de paquetes de R

*Función Necesaria:
update.packages()*

Puesta en marcha de R

R Studio



- Opciones de personalización (apariencia, edición de código)
- Autocompletado
- Atajos de teclado
- Visualización de objetos y tablas
- Comprobación de código en tiempo real, depuración y detección de errores
- Gestión de proyectos



Puesta en marcha de R

Archivos de inicio de R



Los archivos de inicio de R son scripts de R que se ejecutan cada vez que se ejecuta R



Cada línea de estos archivos se ejecuta una vez y solo una vez para cada sesión de R



La modificación de los archivos de inicio de R no debe tomarse a la ligera. Si se modifican los archivos de inicio de manera incorrecta, puede causar problemas



Cuando se inicia R primero busca **.Renviron** y luego **.Rprofile**



Pueden existir múltiples versiones de estos archivos en la misma computadora pero solo una se usa por sesión

Puesta en marcha de R

Archivos de inicio de R

- ❶ **R_HOME:** directorio en el que está instalado R
- ❷ **HOME:** directorio de inicio del usuario (usualmente C:\Users\username en Windows)
- ❸ Directorio de trabajo actual de R

Es importante conocer la ubicación de los archivos de inicio que se desean usar en la sesión. Si hay un archivo **.Rprofile** (o **.Renviron**) en el proyecto actual, R ignorará **.Rprofile** (o **.Renviron**) en **R_HOME** y **HOME**. Del mismo modo, **.Rprofile** (o **.Renviron**) en **HOME** anula **.Rprofile** (o **.Renviron**) en **R_HOME**.

Puesta en marcha de R

Archivos de inicio de R

- 1 **.Rprofile:** Corre líneas de código de R cada vez que R inicia
 - ! Verificar que los paquetes estén actualizados
 - ! Imprimir un mensaje de bienvenida
 - ! Definir funciones que se usan frecuentemente o acortar nombres largos de funciones para mayor eficiencia en la escritura de código

Puesta en marcha de R

Archivos de inicio de R

2

.Renviron: Nos permite crear variables de entorno que le dicen a R donde están ubicados programas externos o que contienen información del usuario que debe mantenerse en secreto



Especificar la ruta donde se deben guardar las librerías de R



Almacenar claves API y tokens de autenticación que estarán disponibles de una sesión a otra



Indicar explícitamente los paquetes a cargar

Consejos y Recomendaciones

Los cinco mejores consejos para una configuración de R eficiente:

- Usar herramientas de monitoreo para identificar cuellos de botella en su hardware/código.
- Mantener actualizada la instalación y los paquetes de R.
- Hacer uso de las potentes capacidades de autocompletado y accesos directos de RStudio.
- Almacenar las claves de API en el archivo .Renviron.
- Considerar la posibilidad de cambiar su biblioteca BLAS.

GRACIAS

Organiza:



**Semillero
Unal**

Colaboran:



**R-Ladies Medellín +
Barranquilla + Galápagos**