

# **manual\_avalor**

Norah Jones

2025-07-29

# Table of contents

<b>Preface</b>	<b>3</b>
<b>1 Preparación del ambiente de trabajo</b>	<b>4</b>
1.1 Como se instala R y Rstudio® . . . . .	4
1.1.1 Descargar e instalar Rstudio . . . . .	5
1.2 Interfaz de Rstudio® . . . . .	6
1.3 Instalación de Paquetes (librerías de funciones) . . . . .	9
1.4 Como instalar y cargar un paquete en R . . . . .	10
<b>2 Comenzar a trabajar con R y la interfaz de Rstudio</b>	<b>15</b>
2.1 Creación de un proyecto . . . . .	15
2.2 Archivos de rutina (script) en R . . . . .	17
<b>3 Introduction</b>	<b>21</b>
<b>4 Summary</b>	<b>22</b>
<b>References</b>	<b>23</b>

# Preface

This is a Quarto book.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

1 + 1

[1] 4

# 1 Preparación del ambiente de trabajo

## 1.1 Como se instala R y Rstudio®

El primer paso para esta tarea es descargar el programa R, que se encuentra en la página web <https://cran.r-project.org/> . en esta hay varias versiones dependiendo que sistema operativo estás usando, ya sea Windows, macOS o Linux.

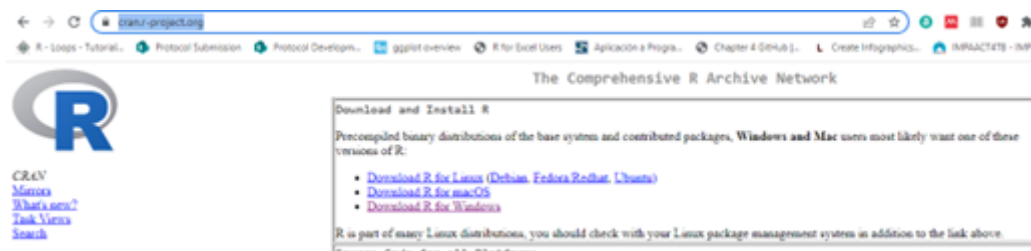


Figure 1.1: En esta página trata de descargar la versión más actualizada

Luego de descargarlo puedes instalarlo inmediatamente usando el ejecutable (para Windows y macOS) o la forma como se instalan software in Linux. Para este manual estamos en un ambiente de Windows, luego de instalar podemos acceder a la **consola** de R (una pantalla con un cursor parpadeando donde podemos escribir)



Figure 1.2: Este es la consola de R, aquí podemos comenzar a usar R, pero es un poco complejo usar solamente la consola

Aunque ya hayamos instalado R, al menos que seamos muy expertos en su uso, que conozcamos bien las sintaxis de las funciones, que *objetos* (más tarde veremos lo que son los objetos) están cargados en la memoria, entre otras cosas (¡también se puede usar como una simple calculadora!), es muy complicado usar solamente la consola, por lo que tenemos disponible aditamentos o accesorios que nos permiten trabajar más fácil con R, así como su aprendizaje. Para este manual el aditamento que usaremos es **Rstudio** de [Posit®](#).

### 1.1.1 Descargar e instalar Rstudio

Ahora vamos a instalar Rstudio, para esto vamos a descargarlo de la siguiente página web, [Descarga de Rstudio](#) donde vamos a buscar la versión gratuita de escritorio para para el sistema operativo que estamos usando.

En caso de que estés usando Linux, están las instrucciones en la página de como hacerlo, la versión de Windows y macOS es un ejecutable.

Luego de descargar la ultima versión disponible procedemos a instalar Rstudio, haciendo clic en el ejecutable, la instalación (en la versión de Windows por ejemplo) es muy similar a cualquier otro software que donde te pregunta el lugar donde será instalado y varias ventanas donde se ve el progreso de instalación. Si todo salió bien, es decir que se instaló sin errores, pues tendremos disponible en la barra de acceso directo y en el escritorio, (si elegimos esta opción durante la instalación) también tendremos un acceso directo.

Si tienes Windows, te compartimos el paso a paso en [este video](#) para descargar e instalar R y también R Studio Desktop.

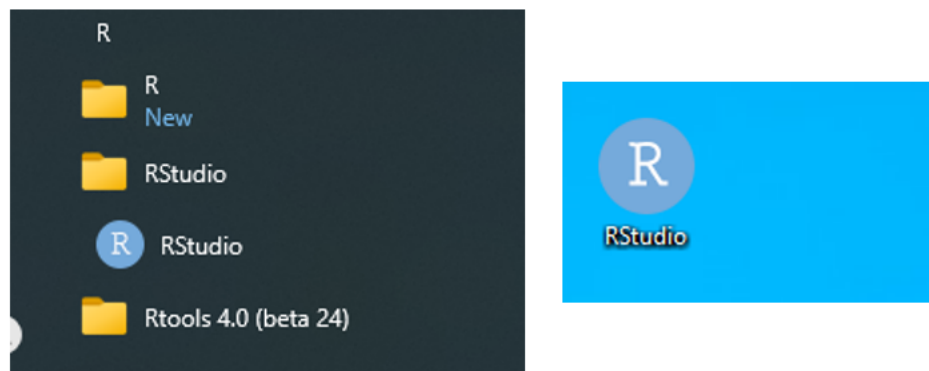


Figure 1.3: En Windows, la barra de inicio o las aplicaciones y si se creó el acceso directo (derecha) en el escritorio

Otro software que se debe instalar si el sistema operativo que usas es Windows es **rtools**, dado que algunos paquetes necesitan que esté instalado para funcionar de forma correcta.

Este se adquiere desde la página de [Descarga rtools](#) y elige la versión más reciente y descarga el formato ejecutable (termina en “installer”) en cualquier unidad de almacenamiento. Su instalación sencilla, solo ejecutar el programa y hacer clic en siguiente las veces que sea necesario.

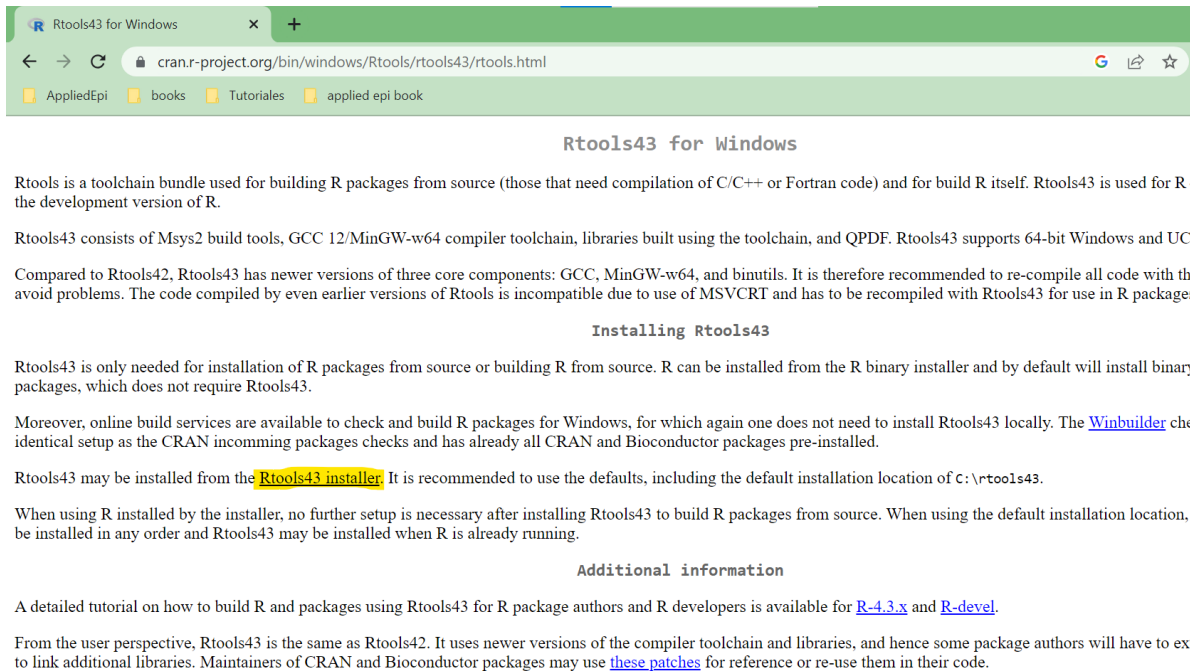


Figure 1.4: Busca esta version señalada en la imagen

## 1.2 Interfaz de Rstudio®

Para facilitar el aprendizaje y uso de R, vamos a usar RStudio que es un entorno de desarrollo integrado o IDE (Integrated Development Environment) y tiene la gran ventaja de que hay mucha documentación sobre su uso, es muy cómoda de trabajar porque nos ayuda con la escritura de los códigos, la organización de los archivos, tiene varios visores o paneles para ver los códigos, las salidas, los objetos cargados y otras ayudas más (Figura 5). RStudio ha permitido la difusión del uso de R, lo que ha permitido la propagación de su uso en todas las áreas donde se hace análisis de datos.

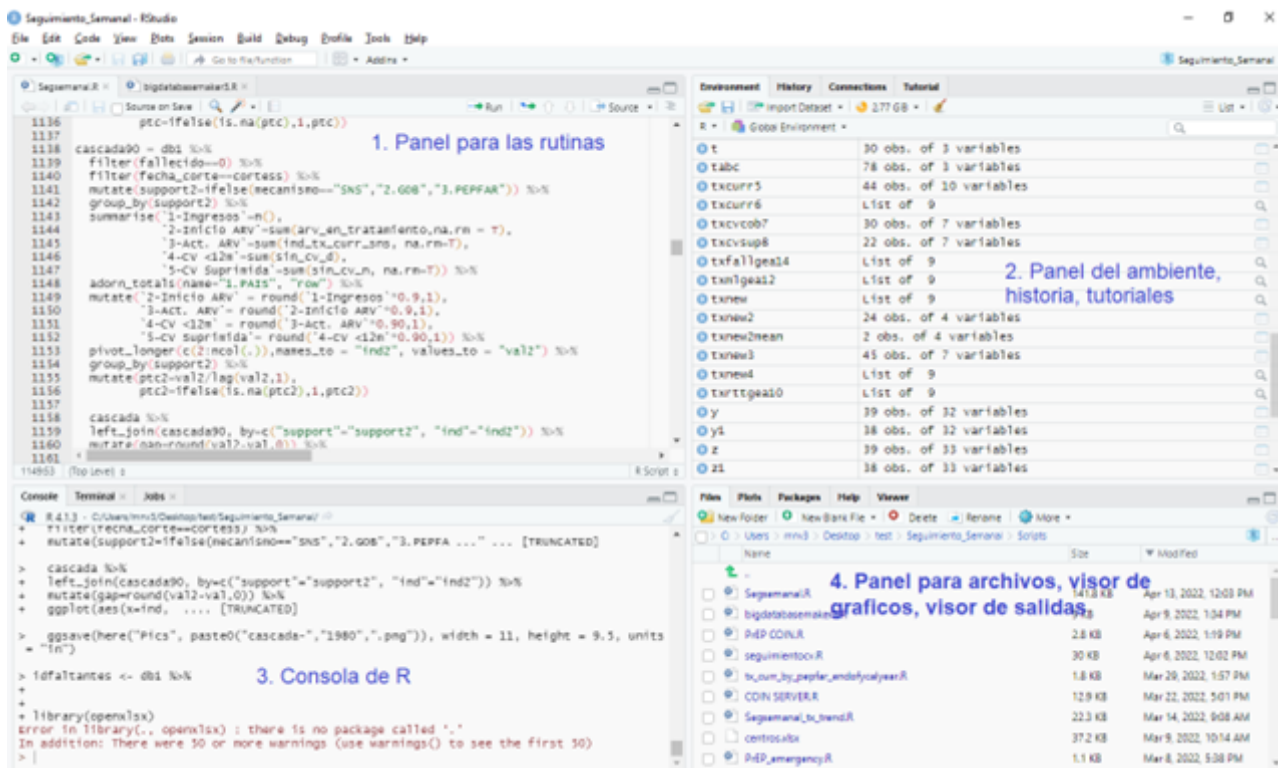


Figure 1.5: En esta imagen podemos ver los 4 paneles principales y el menú de la interfaz de Rstudio

- 1. Panel de las rutinas:** este panel es el editor de texto donde vamos a escribir los códigos (rutinas) que vamos a usar para nuestras tareas, como las tablas, los gráficos, etc., que son automáticamente ejecutadas en la consola de R. Cada rutina se puede guardar como un archivo (muy similar a un archivo de texto normal), la característica más importante de este panel es que nos permite ver si el código tiene errores, pues a veces la falta de una simple coma o un caracter nos arroja error cuando ejecutamos comandos. También nos ofrece la herramienta de autocompletar y nos facilita mucho la organización del código. Para ejecutar una línea de código simplemente ponemos el cursor al inicio o al final de la línea y presionamos Ctrl+Enter o en el botón “run”, mientras que para ejecutar la rutina completa, hacemos clic en el botón “source”. Más adelante veremos varios ejemplos.
- 2. Panel del ambiente de trabajo o panel de objetos cargados:** en este panel podemos ver cuales objetos tenemos cargados en la memoria del sistema, y nos permite ver qué tipo de objeto es (si es un *dataframe*, un *vector*, una *matriz*, una *función*, una *lista*). Esto nos ayuda a seguir el trabajo que vamos realizando, por ejemplo, cuando cargamos una base de datos desde un archivo de Excel en un objeto *dataframe* podemos ver cuantas filas y variables tiene el archivo. También a través de este panel podemos salvar la sesión de trabajo, esto es útil cuando hacemos una pausa y queremos retomar más adelante lo

que estábamos haciendo.

3. **Consola de R:** este es el lugar donde ocurre todo, puedes directamente escribir los comandos, las funciones, etc., pero para esto tenemos el panel de las rutinas, aquí también vas a ver los mensajes de errores cuando ejecutas una rutina o un comando, y por cada línea que se escribe se presiona “enter” para ejecutar los comandos. La ventaja más grande de utilizar RStudio es que hasta en la consola te identifica si hay errores o comandos incompletos y autocompletar.
4. **Panel para archivos, visor de gráficos y de salidas:** en este panel tenemos varias ventanas donde nos permite ver los archivos disponibles (muy parecido al explorador de Windows o a la carpeta de Documentos de tu computadora). A través del menú de este panel podemos crear carpetas, borrar, mover o copiar archivos, también podemos definir el directorio de trabajo (vamos a ver más adelante en detalle qué es el directorio o lugar de trabajo), a la derecha de Archivos o Files está el visor de gráficos (Plots), que podemos ampliar y poder copiar el gráfico que se está presentando y más a la derecha está el visor de salidas (Viewer), como tablas y datos en formato HTML. También en este panel está la ventana para instalar paquetes (término muy importante que lo veremos más adelante) y la ventana de ayuda (Figura 6).

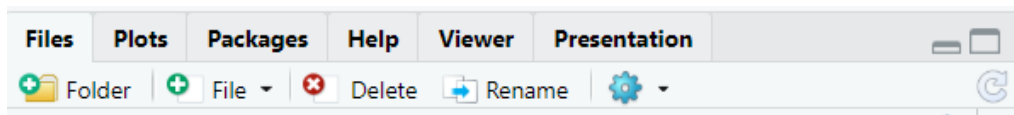


Figure 1.6: Figura 6.

Antes de comenzar a trabajar puedes ir familiarizándote con esta interface. En RStudio hay muchas opciones que iremos explicando en la medida que vayamos avanzando, mientras tanto, te mostrados dos herramientas interesantes del menú principal de RStudio: **Tools** y **Help**.

Si vas al menú principal de RStudio, puedes entrar en la sección que dice “**Tools**” y dentro del menú desplegable seleccionas la opción de “**Global Options**” (al final del menú desplegable), allí encontrarás un menú, y dentro de “**Appearance**” está la opción “Editor theme” donde podrás cambiar los colores y el tipo de letra de las ventanas, así podrás para adaptar el ambiente a tu gusto. En Ayuda (Help) puedes encontrar los accesos directos para usar el teclado, por ejemplo, salvar la rutina en la que estás trabajando, reiniciar R, ejecutar toda la rutina, entre otros atajos. En la Figura 7 te mostramos cómo llegar a cada una de estas herramientas.



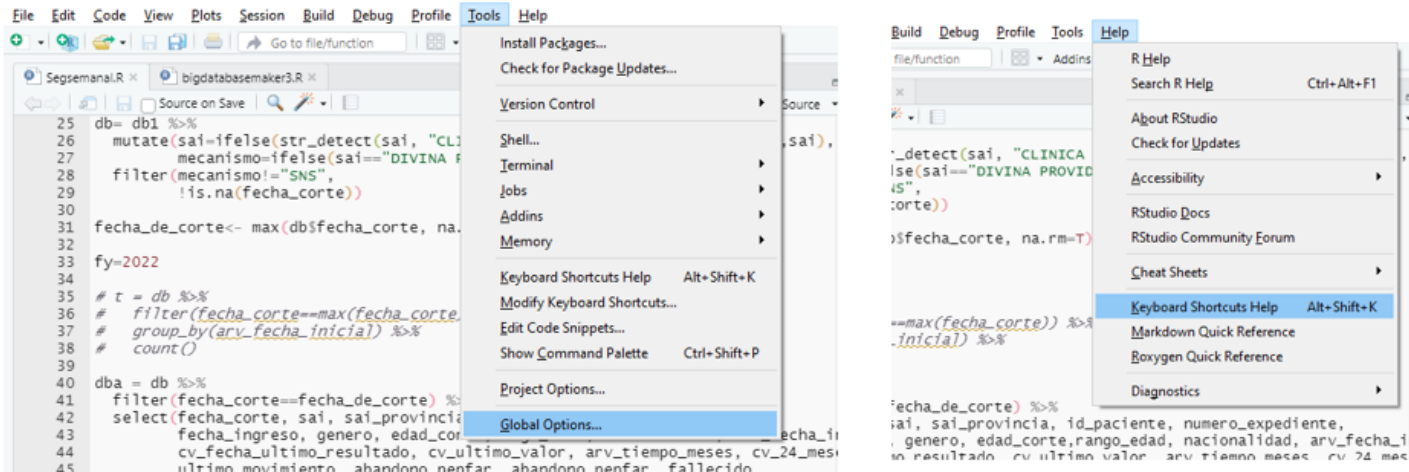
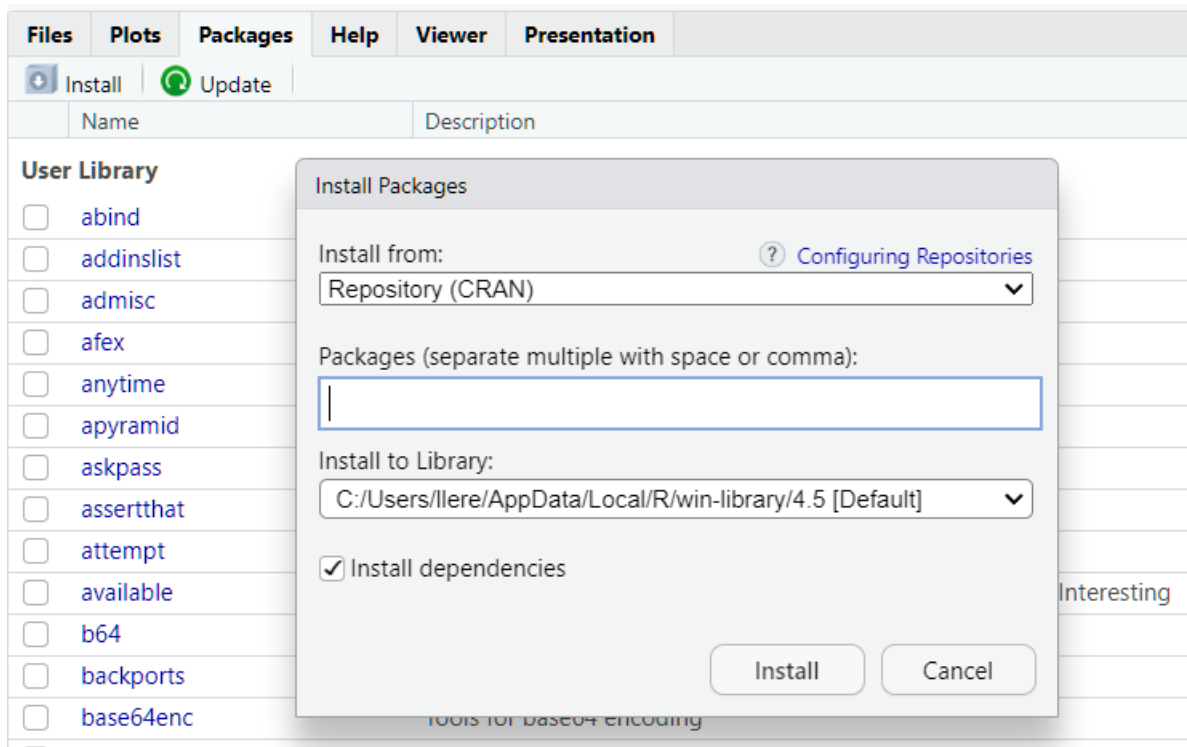


Figure 1.7: Figura 7. Seleccionar “Tool” luego “Global Options” y en ayuda, buscar los atajos del teclado

### 1.3 Instalación de Paquetes (librerías de funciones)

Los paquetes son extensiones para añadir más funcionalidades a R. Es quizás la razón por la cual R es ahora mismo una de las mejores herramientas para trabajar con datos dada la cantidad de paquetes específicos para realizar tareas como hacer tablas por ejemplo. En la literatura, los paquetes se identifican con { }, ejemplo: **{tidyverse}** y los logos son una figura dentro de un hexágono.

Con respecto a la instalación de paquetes, usualmente estos deben estar publicados en CRAN ([Aquí el enlace](#)) y se instalan desde el panel de paquetes (“Packages”).



## 1.4 Como instalar y cargar un paquete en R

Después de instalar R, Rstudio solo tenemos las funcionalidades que trae **R base**, que nos permiten hacer muchas cosas sin tener que instalar nada más, pero podemos hacer que R sea más completo, ahora vamos a ver que es un paquete y su importancia.

Como R es un lenguaje de programación por lo tanto podemos crear programas o conjunto de funciones y estos programas podemos exportarlos para luego usarlos más adelante.

Estos programas o complementos aumentan la capacidad de R a través de funciones o comandos, por ejemplo, **R base**, no nos permite exportar directamente los resultados o salidas en formato Excel o crear tablas con formato de forma directa. Por lo tanto, los paquetes o subprogramas hacen que R sea más que solo un lenguaje de programación.

Hay paquetes creados por epidemiólogos, para epidemiólogos como el **epitools** (Tomas J. Aragon) ([aragon2020?](#)) o **epikit** (Zhian N. Kamvar)([spina2023?](#)) que nos permiten con pocos comando o funciones hacer tareas como calcular medidas de asociación (OR, Riesgo relativo) y facilitar los análisis en epidemiología y otros paquetes que hacen las tareas de análisis de datos como son el **openxlsx** (Philipp Schaubberger)([schaubberger2023?](#)) para manipular y crear archivos de Excel o el muy famoso y aclamado paquete **tidyverse** (Hadley

Wickham)(wickham2023?)del cual hay este paquete hay libros escritos para el procesamiento y manejo de datos por mencionar algunos.

Los paquetes nos permiten mejorar sobremanera nuestra productividad en general. Cada vez que iniciamos una sesión con Rstudio, en nuestras rutinas debemos cargar los paquetes que vamos a usar.

Dado que R es una plataforma colaborativa, cuando un paquete es creado pasa por un proceso de validación antes de ser publicados en el repositorio de R. Es por esto que, para su instalación debemos saber si están disponibles. Los paquetes, usualmente, después de ya ser validados, son publicados en el repositorio de R (The Comprehensive R Archive Network o [CRAN](#)) que es la página donde descargamos R. También pueden estar disponibles en otras páginas donde podemos descargarlos e instalarlos manualmente. Después de instalar un paquete solo es necesario verificar si ha sido actualizado, tal como haces con cualquier otro programa.

Usualmente los autores de los paquetes tienen tutoriales o páginas llamadas “Vignettes” o viñetas donde podemos ver las funcionalidades y tutoriales. Por ejemplo, ejemplo ingresa en la barra de búsqueda en tu explorador estas palabras: “R package epitools”, y aparecerán muchos resultados para diferentes tareas. Sigue estos pasos para verificar si un paquete existe buscando directamente desde RStudio:

1. Ir al panel de archivos
2. Hacer clic en la ventana de paquetes o “Packages”
3. Hacer clic en “install”. Aparecerá una ventana donde escribirás el nombre del paquete.  
Si el paquete existe en CRAN, aparecerá en un listado y lo seleccionarás
4. Hacer clic en “install”.

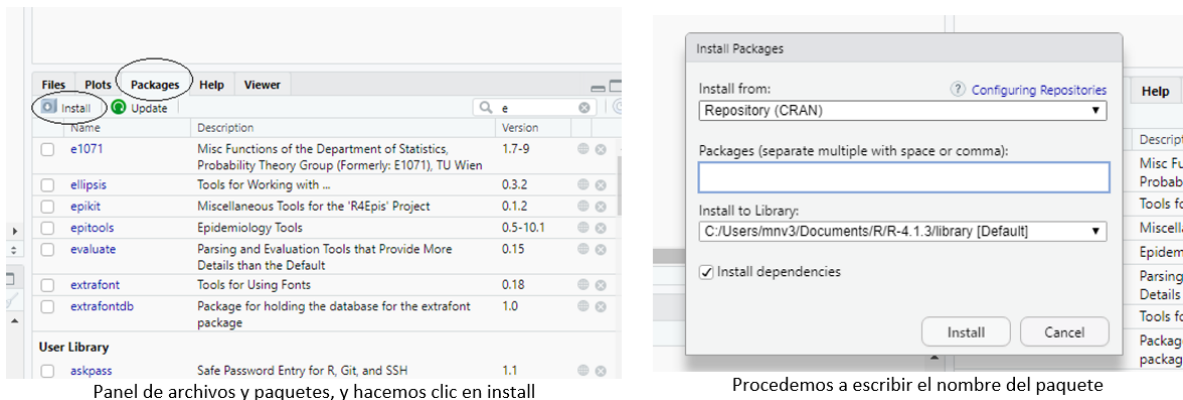


Figure 1.8: installing a package in Rstudio

Si el paquete no está disponible en CRAN pero si en la web y podemos descargarlo y cambiar donde dice “install from” a “package archive file” y buscar en el disco duro y proceder

a instalar (es muy raro que se de este caso y los paquetes que vamos a instalar todos están disponible en CRAN).

La otra forma de instalar los paquetes es directamente desde la consola o en un archivo de rutina usando el la función `install.package()` donde escribimos entre comillas el nombre del paquete que queremos instalar.

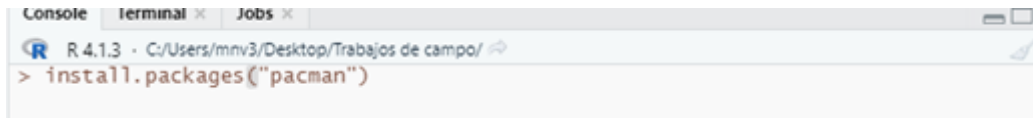


Figure 1.9: Figura 14. En esta imagen vemos en la consola la función para instalar el paquete pacman

Luego de esta introducción sobre qué es un paquete y cómo se instala, vamos a hacer el siguiente ejercicio de instalar varios paquetes que usaremos de forma constante en los ejercicios y tareas contenidos en este manual. Ya sea directamente en la consola (ver imagen anterior), o a través del panel de archivos, vamos a instalar este paquete “**pacman**” (Tyler Rinker) que es un paquete para manejar paquetes y nos ahorrará muchos pasos, por ejemplo, detecta si un paquete necesario está instalado o no, y procede a instalarlo, o ayuda a instalar paquetes desde otras fuentes alternativas a CRAN. Después de instalarlo podemos ver en la consola el mensaje de que se instaló correctamente (*package ‘pacman’ successfully unpacked and MD5 sums checked*).

En la rutina que comenzamos hace un momento atrás vamos a hacer la siguiente tarea:

- Escribe o copia y pega el siguiente comando (recuerda, para ejecutar un comando o varios selecciona estos y presiona Ctrl+Enter o clic en “Run”):

```
pacman::p_load("tidyverse", "janitor", "gtsummary", "epikit", "here",
```

```
"epitool"
```

Espera un momento si es la primera vez que se ejecuta para que así se instalen los paquetes que están en el comando.. Antes de continuar, vamos a explicar el comando anterior:

En Rstudio, en el editor de rutinas o en la misma consola, cuando queremos ver las funciones o comandos disponibles en un paquete podemos escribir el nombre del paquete seguido de dos puntos nos aparecerá una ventana con un listado de dichas funciones, por ejemplo, este último comando usamos el paquete **pacman** y dentro de este paquete usamos la función de `p_load` (para cargar paquetes, que también los instala si no están ya instalados)

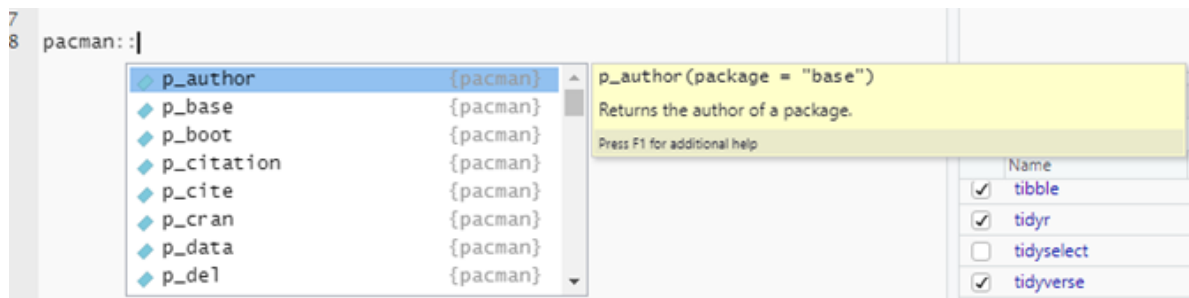


Figure 1.10: Esta es una de las funcionalidades de Rstudio que nos facilitan el trabajo , luego de escribir el nombre del paquete y dos veces dos puntos nos aparece el listado de funciones y también una ventana amarilla para ayuda de la función presionando F1 para ver en el panel de archivos, la ventana de ayuda y así ver los ejemplos de cómo usar dicha función.

Luego de escribir el nombre del paquete y dos veces dos puntos, nos aparece el listado de funciones y también una ventana amarilla para ayuda de la función, que también aparece presionando F1 o haciendo una búsqueda en el panel de archivos en la ventana de ayuda, donde también se pueden ver los ejemplos de cómo usar las funciones.

Los paquetes que hemos instalado hasta este momento no los vamos a detallar por ahora, sino que, en la medida que vayamos haciendo los ejercicios, vamos a ir explicando para qué sirven a través de las funciones que traen cada uno. Es oportuno señalar que, dentro de los paquetes disponibles en R, existen paquetes útiles para facilitar el proceso de importar y exportar funciones específicas para las tareas del epidemiólogo, para hacer reportes y para facilitar el proceso de la gestión de los datos.

Para ver que paquetes tenemos cargados podemos escribir en la consola `search()`

```
search()
```

```
[1] ".GlobalEnv"      "package:rio"      "package:readxl"
[4] "package:openxlsx" "package:epitools" "package:here"
[7] "package:epikit"   "package:gtsummary" "package:janitor"
[10] "package:lubridate" "package:forcats"   "package:stringr"
[13] "package:dplyr"     "package:purrr"     "package:readr"
[16] "package:tidyr"     "package:tibble"    "package:ggplot2"
[19] "package:tidyverse" "package:stats"     "package:graphics"
[22] "package:grDevices" "package:utils"     "package:datasets"
[25] "package:methods"  "Autoloads"         "package:base"
```

Los dos capítulos siguientes son muy importantes. Debes familiarizarte con ellos porque son los fundamentos para poder trabajar en R, veremos ejemplos de las sintaxis de las expresiones,

los diferentes tipos de objetos de datos, y a medida que vayamos practicando se irá haciendo más fácil entender el código, las funciones y las operaciones que hacemos. Te recomendamos que, aparte de este manual, abundes más sobre los temas que verás a continuación para que agilices tu proceso de aprendizaje con R.

## 2 Comenzar a trabajar con R y la interfaz de Rstudio

Ya luego de tener todo lo necesario instalado, ahora vamos a realizar los siguientes pasos con la finalidad de ir aprendiendo a organizar los trabajos que estaremos haciendo, como son los productos o asignaciones que se deben hacer durante la capacitación, como son los análisis de vigilancia, análisis de brotes, evaluación de sistema de vigilancia entre otros.

Una gran recomendación es tener lo más organizado posible cada elemento hecho en R o cualquier documento asociado a los análisis o trabajos que vas a realizar; esto porque en nuestro caso, cuando comenzamos a trabajar con RStudio guardábamos las rutinas y las salidas en diferentes lugares, como principiantes al fin, no teníamos una estructura, y luego cuando necesitábamos re-usar una rutina, buscar un documento de salida era muy difícil de encontrar y perdíamos tiempo en esa búsqueda, a veces nos tomaba más que lo que se toma ejecutar una rutina. Luego aprendimos sobre lo que vamos a ver a continuación en cuanto a trabajar bajo proyectos en RStudio.

### 2.1 Creación de un proyecto

El primer paso para comenzar es decidir dónde estarán los archivos relacionados al o a los trabajos que se van a hacer, es decir, debes definir si almacenarás tu trabajo en un disco dentro de la PC o en la “Nube” (ejemplo, OneDrive, Google drive) para que así siempre facilites tu trabajo, especialmente cuando necesites hacer cambios o actualizar las bases de datos.

Esto es sumamente importante dado que nos ayudará a mantenernos organizados. Todos los archivos como las bases de datos, las referencias, las rutinas, deben estar almacenados en un lugar donde te sea fácil encontrarlos y también para la ejecución o desarrollo del proyecto o proyectos en que estés trabajando.

Para el segundo paso vamos a entrar en RStudio y vamos a crear un **archivo de proyecto** de la siguiente forma:

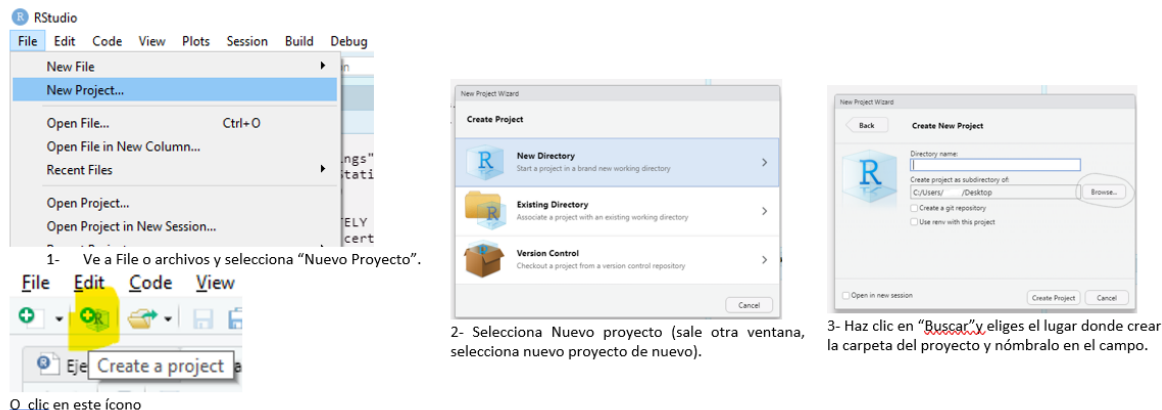


Figure 2.1: Figura 10. Creando un proyecto

1. Crea el proyecto llamándolo "mi\_primer\_proyecto" siguiendo los pasos de la Figura 10.
2. Guarda el proyecto nuevo en el "escritorio" o en "Documentos" si usas Windows.
3. Para crear las carpetas, en la ventana inferior derecha o panel para archivos, ve a "Archivos" o "File", luego haz clic en "new folder" para crear las siguientes carpetas:
  - a. "Base de datos" para almacenar tus datos en cualquier formato, como datos en Excel, por ejemplo.
  - b. "Rutinas" para salvar las rutinas que irás creando.
  - c. "Referencias" vas a poner aquí los documentos de soporte como referencias, artículos, etc.
  - d. "Salidas", donde vamos a salvar los documentos generados a partir de las rutinas, en esta última podemos tener una subcarpeta que se llame "imágenes" para las salidas que son imágenes, como gráficos o tablas salvadas en formato de imagen.



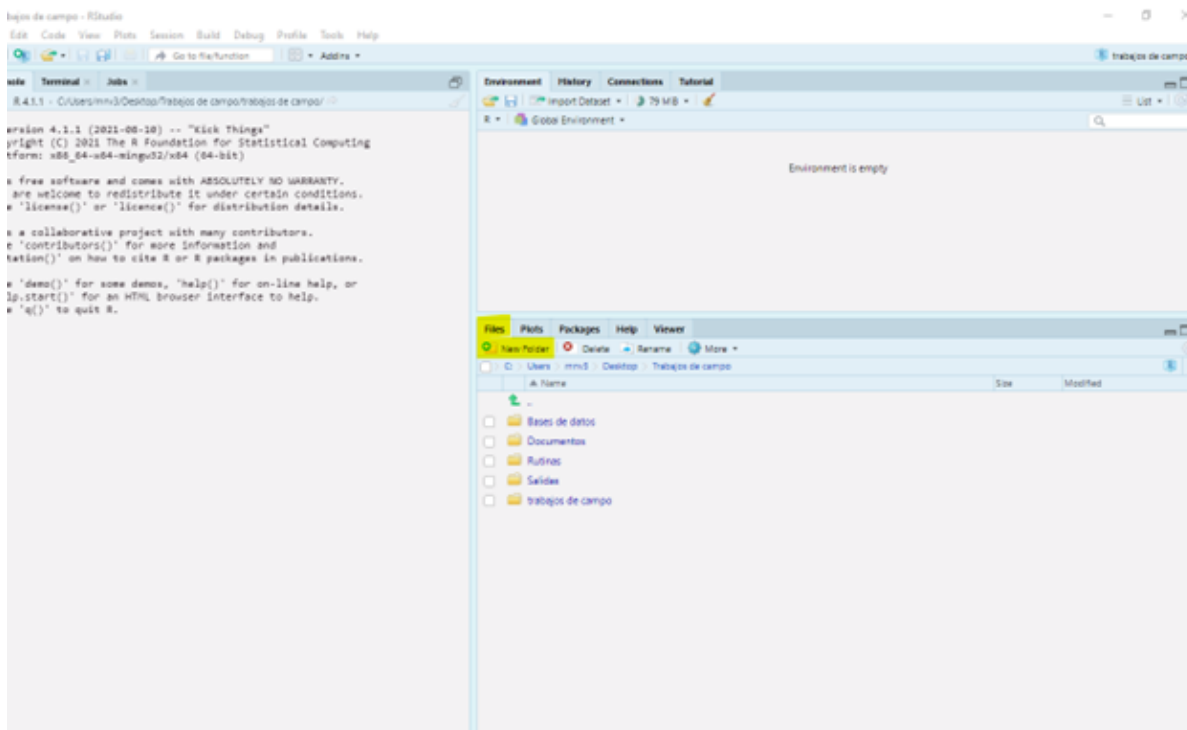


Figure 2.2: Al finalizar, tu panel de archivos se tendría que ver como se muestra en la Figura 11

Si has hecho todos estos pasos y tu pantalla se parece a la imagen anterior es porque has creado un proyecto. Puede verificar el nombre del proyecto en la esquina superior derecha de la interfaz de RStudio.

Aparte de mantener una organización para mantener todo en orden, la mayor ventaja de usar proyectos es que a RStudio se le facilita encontrar dónde están los archivos que serán usados, por ejemplo, si ubicas las bases de datos correctamente, será más fácil para ti importar los datos a R y también exportar archivos a otras herramientas de análisis.

### *¿Por qué en la imagen anterior falta un panel*

En la imagen anterior solo vemos 3 ventanas porque no hemos abierto o creado una rutina, solo tenemos la consola, el panel del ambiente de trabajo y el panel de archivos.

## 2.2 Archivos de rutina (script) en R

Siguiendo la secuencia de pasos para comenzar a trabajar con R en Rstudio, vamos a ver uno de los elementos más importantes que son las rutinas y que la usaremos constantemente.

Hasta este punto, hemos mencionado varias veces la palabra “rutina”. Una rutina no es más que un “documento” donde escribimos comandos (funciones, cálculos, etc.), que puede ser ejecutado las veces que sea necesario (como por ejemplo procesar y generar un reporte semanal). Realmente crear una rutina es un procedimiento muy sencillo porque se trata simplemente de escribir en un editor de texto las funciones que generan los resultados que esperamos.

Dado que podemos salvar las rutinas como archivos, podemos compartirlas y guardarlas para luego usarlas como referencia.

Para crear una nueva rutina en RStudio vamos a “file” -> “new file” -> “R Script” (en inglés rutina es igual a *Script* por lo que estaremos utilizando ambos conceptos de manera indistinta a lo largo del curso).

Después de aceptar ahora tenemos el panel de rutinas habilitado (igual que un editor de texto) tal como se ilustra en la Figura 12.

Cuando salves la rutina nueva por primera vez, te pedirá dónde guardar el archivo, entonces procede a guardarlo en la carpeta de “Rutinas” que previamente creaste en el Panel de archivos y ya podrás comenzar a trabajar. Es importante anotar que RStudio no salva el avance de tu rutina automáticamente, sino que siempre debes guardar cada cierto tiempo, ya sea a través del menú o utilizando el atajo **Ctrl+S**.

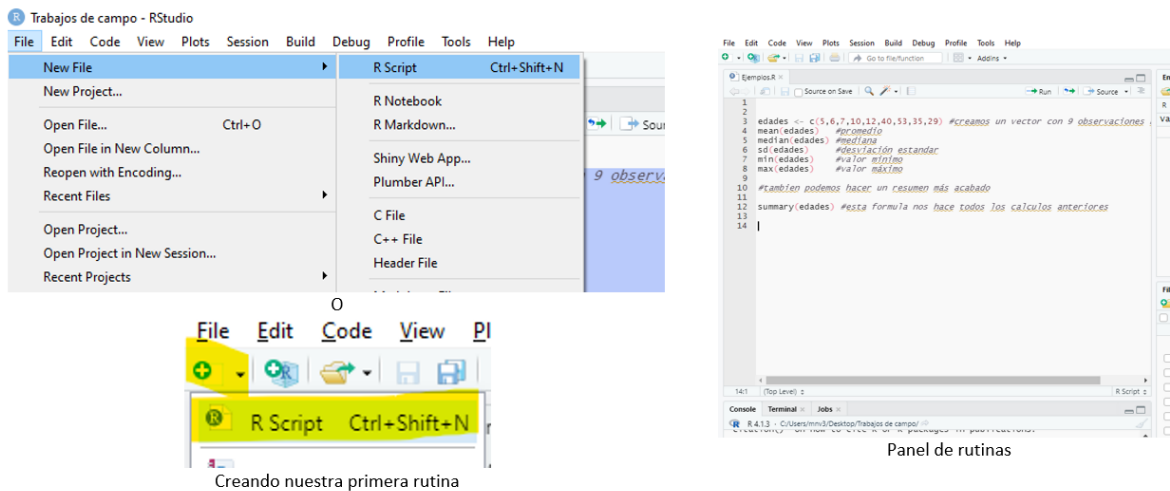


Figure 2.3: Figura 12. Pasos para comenzar un archivo de rutinas o (Ctrl+Shift+N)

Si has llegado hasta aquí, entonces ya tienes una gran parte del camino recorrido, es decir, ya tienes el programa instalado y disponible, un proyecto creado y tu primera rutina guardada. Si desde ya comenzáramos a trabajar, podemos escribir comandos directamente en la consola o en el documento de la rutina para hacer operaciones como aplicar a nuestros datos los

**estadísticos de tendencia central**, para que vayas viendo y familiarizándote con R. Vamos a hacer la siguiente prueba paso a paso:

1. Escribe el texto debajo en el documento de la rutina, o si estas desde tu PC, copia y pega todo este texto, selecciónalo (como en cualquier editor de texto o Ctrl+A) y presiona **Ctrl+Enter** o haz clic en “**Run**”.

```
edades <- c(5,6,7,10,12,40,53,35,29) #creamos un vector con 9 observaciones numericas
mean(edades)    #promedio
median(edades)  #mediana
sd(edades)      #desviación estandar
min(edades)     #valor minimo
max(edades)     #valor máximo
#tambien podemos hacer un resumen más acabado, con menos comandos
summary(edades) #esta función nos hace todos los calculos anteriores}
```

2. Luego de ejecutar los comandos, podemos ver el resultado en la consola. Como puedes observar, está escrito lo mismo que la rutina pero debajo de cada comando hay un resultado, es decir, vas a ver el resultado de calcular el promedio a través de la función mean() cuyo resultado es 21.8, y así sucesivamente. La consola debería verse algo similar a esto:

```
edades <- c(5,6,7,10,12,40,53,35,29) #creamos un vector con 9 observaciones numericas
mean(edades)    #promedio
```

```
[1] 21.88889
```

```
median(edades) #mediana
```

```
[1] 12
```

```
sd(edades)      #desviación estandar
```

```
[1] 17.73728
```

```
min(edades)     #valor minimo
```

```
[1] 5
```

```
max(edades)      #valor máximo
```

```
[1] 53
```

```
#tambien podemos hacer un resumen más acabado, con menos comandos  
summary(edades) #esta función nos hace todos los calculos anteriores
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
5.00	7.00	12.00	21.89	35.00	53.00

Una de las ventajas que tiene RStudio de trabajar con los documentos de rutina es que puedes tener varias rutinas abiertas a la misma vez y navegar entre ellas. Esto te puede resultar útil para utilizar una rutina dentro de otra, es decir, como las rutinas se pueden guardar en la PC, vas a darte cuenta de que hay muchos procedimientos que se repiten y que solo necesitas modificarlos levemente, en estos casos, mientras estás haciendo una rutina puedes usar otra como referencia para tomar códigos que ya has usado antes o que te han compartido. *¿Cuál es la ventaja de esto? ¿que no es obligatorio aprenderse todas las funciones o comandos de R!*

Una muy buena práctica es comentar todo lo que haces para luego saber qué hiciste en una secuencia de códigos. Para esto, solo tienes que comenzar una línea con el carácter “#” o signo de número, y todo el texto después de este carácter cambia de color y se pone en formato itálico, tornando el texto de color verde de forma predeterminada, aunque el color verde pudiera variar dependiendo de la configuración de colores que hayas elegido, tal como te explicamos en la sección 2.2. Cuando estemos en los pasos de hacer tablas y gráficos, vamos a usar mucho los comentarios.

El editor de texto de rutina en RStudio también nos permite ver si hay errores en el código, otra de las tantas funcionalidades que nos ofrece el programa para hacer más fácil la escritura de códigos.

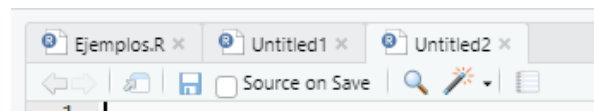


Figure 2.4: Varias rutinas abiertas (con **Ctrl+shift+tab** podemos movernos entre las rutinas sin usar el ratón)

## 3 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

## 4 Summary

In summary, this book has no content whatsoever.

1 + 1

[1] 2

## References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.