

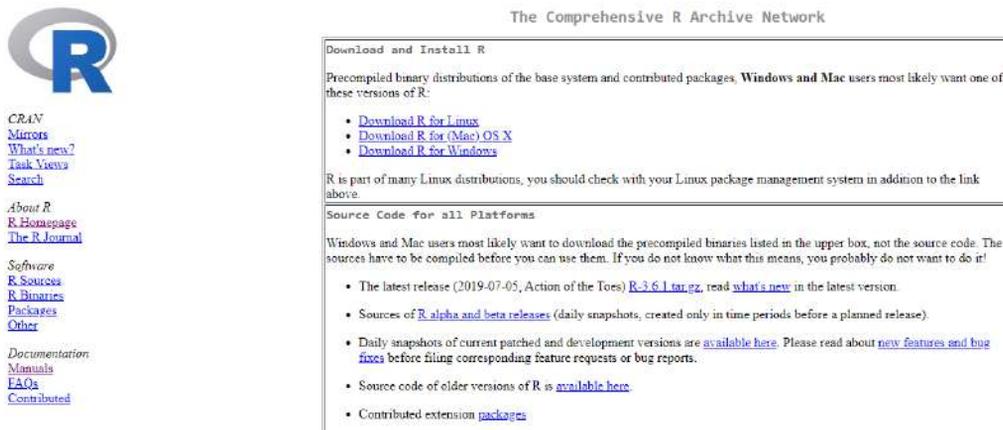
## Introducción al lenguaje R

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a estos considera que debes apegarte a sus términos y condiciones.

### Instalación de R

Para instalar R accede a la siguiente página: <https://cran.r-project.org/>.

Una vez en la página, selecciona el sistema operativo con el que vas a trabajar (dependerá del sistema operativo que tengas instalado en tu computadora). En este caso, supongamos que seleccionamos el de Windows:



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](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

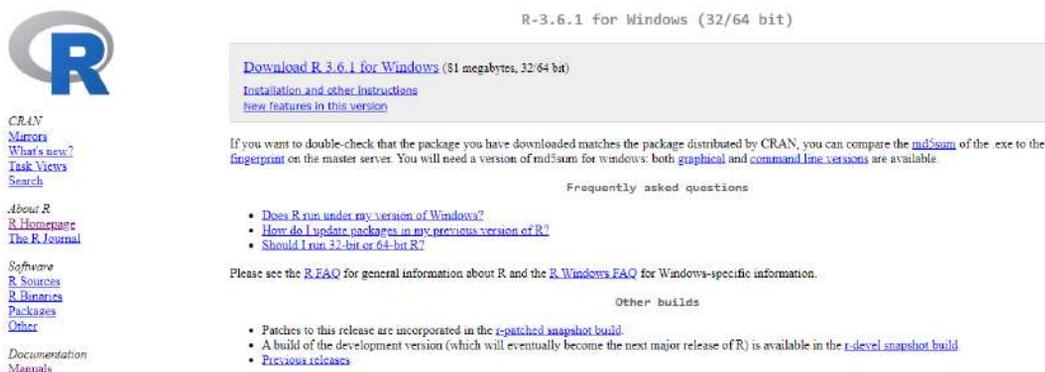
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

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 (2019-07-05, Action of the Toes) [R 3.6.1 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](#).

Después, selecciona la opción de base (que implica instalar R por primera vez) para que te traslade a la siguiente página, en donde seleccionarás la versión que se te indique en primer plano, ya que esta será la versión más actualizada. Haz clic en esa opción aunque no coincida con la que aparece en la siguiente imagen. Se comenzará a descargar un archivo que te guiará en el proceso de instalación:



R-3.6.1 for Windows (32/64 bit)

[Download R 3.6.1 for Windows](#) (81 megabytes, 32/64 bit)

[Installation and other instructions](#)  
[new features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of [md5sum](#) for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Sigue los pasos aceptando la opción de instalación que viene por *default* hasta que llegues a la opción de Finalizar:



Prosigue a instalar RStudio, que es un entorno de desarrollo integrado (IDE), el cual provee de un editor que simplifica la programación en este lenguaje. Para descargarlo puedes dirigirte a la siguiente página: <https://rstudio.com/>

Una vez ahí, selecciona descargar RStudio en su versión gratuita de escritorio. Posteriormente, aparecerá una página con todas las opciones de instalación dependiendo del sistema operativo. Es importante coincidir con el sistema operativo seleccionado en la instalación previa de R:

### RStudio Desktop 1.2.5019 - Release Notes

1. Install R. RStudio requires R 3.0.1+.
2. Download RStudio Desktop. Recommended for your system:



Requires Windows 10/8/7 (64-bit)



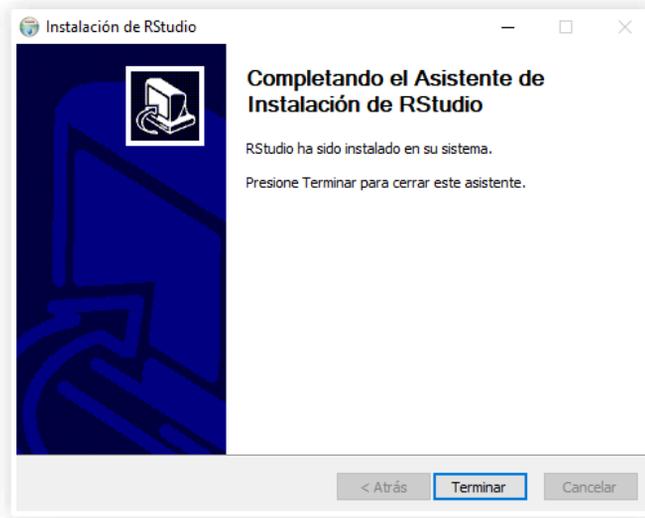
### All Installers

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

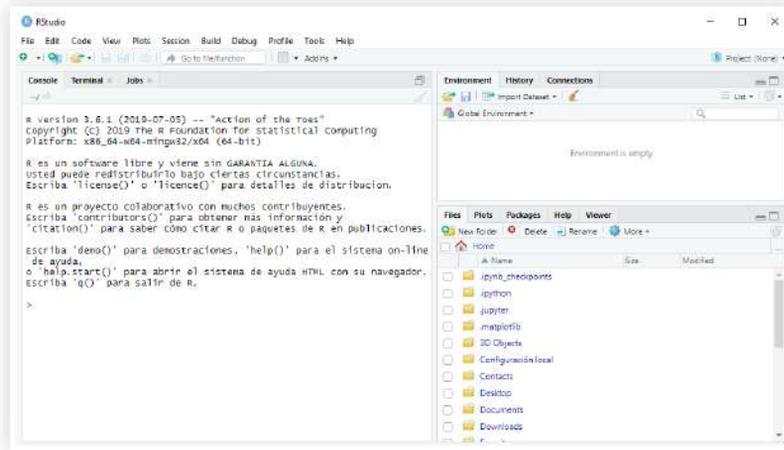
RStudio 1.2 requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.

OS	Download	Size	SHA-256
Windows 10/8/7	<a href="#">RStudio-1.2.5019.exe</a>	149.82 MB	7c5e943c
macOS 10.12+	<a href="#">RStudio-1.2.5019.dmg</a>	120.00 MB	00c17d64
Ubuntu 14/Debian 8	<a href="#">rstudio-1.2.5019-amd64.deb</a>	96.93 MB	a9f43062
Ubuntu 16	<a href="#">rstudio-1.2.5019-amd64.deb</a>	104.91 MB	28Feb367

A manera de ejemplo, en este laboratorio se seleccionará la versión para Windows, en la cual se seguirán los pasos del asistente hasta terminar la instalación:

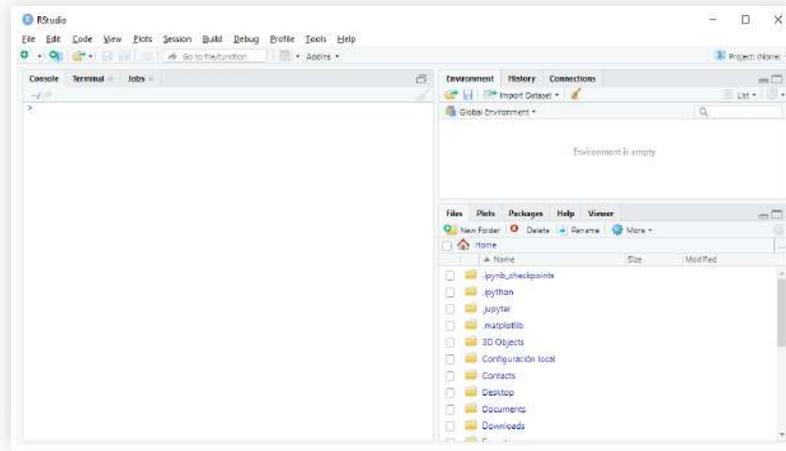


Inicia la aplicación de Rstudio y haz clic para que te lleve a la ventana principal:

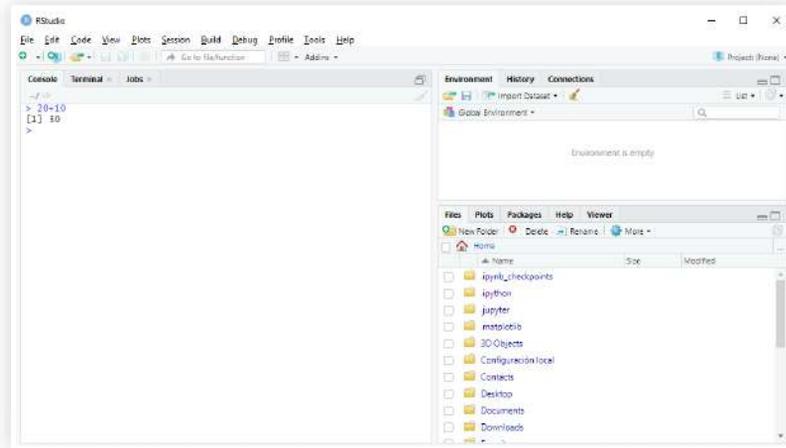


## Operaciones aritméticas básicas

Presionando las teclas Ctrl+L podemos limpiar la pantalla de la consola:



Las operaciones aritméticas básicas las puedes realizar directamente, por ejemplo, si deseas sumar dos números como  $10+20$ , se escribiría la operación tal cual en el prompt:



En el caso de multiplicación:

```

> 30*50
[1] 1500

```

División:

```

> 100/5
[1] 20
>

```

Agrupación de operaciones:

```
> 5*(4+6)
[1] 50
>
```

R también cuenta con una gran variedad de funciones, por ejemplo, si defines un vector de datos puedes obtener el máximo y el mínimo con funciones, como se muestra a continuación:

```
> x<- 1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10
> max(x)
[1] 10
> min(x)
[1] 1
```

El operador `<-` sirve para asignar valores a variables. Por otra parte, una forma de crear vectores es con la función `combine c()`, donde los elementos deben incluirse separados por coma, por ejemplo:

```
> y<- c(5, 10, 15, 20, 25, 30, 35, 40, 45, 50)
> y
[1] 5 10 15 20 25 30 35 40 45 50
```

A este vector que definimos como **y** podemos realizarle ciertas operaciones con las funciones, por ejemplo, sumar sus elementos:

```
> sum(y)
[1] 275
```

Multiplicarlos:

```
> prod(y)
[1] 3.54375e+13
```

Funciones trigonométricas:

```
> sin(y)
[1] -0.9589243 -0.5440211 0.6502878 0.9129453 -0.1323518
[6] -0.9880316 -0.4281827 0.7451132 0.8509035 -0.2623749
> cos(y)
[1] 0.2836622 -0.8390715 -0.7596879 0.4080821 0.9912028
[6] 0.1542514 -0.9036922 -0.6669381 0.5253220 0.9649660
```

Así como algunas operaciones de estadística descriptiva:

```
> mean(y)
[1] 27.5
> median(y)
[1] 27.5
> sd(y)
[1] 15.13825
```

También se puede asignar texto a las variables, por ejemplo:

```
> a <- "Buen día"
> a
[1] "Buen día"
>
```

De igual forma se puede usar la función `combine c()` para hacer una combinación de textos dentro de un conjunto:

```
> b <- c("Buen día", "todos")
> b
[1] "Buen día" "todos"
```

En caso de que se desee que el usuario de un programa pueda introducir comentarios o respuestas a preguntas se puede utilizar la función `readline()`, que sirve para leer una línea en la terminal, por ejemplo:

```
> x <- "hola"
> edad <- readline("¿Cuántos años tienes? ")
¿Cuántos años tienes? 21
> edad
[1] "21"
```

**Ahora es tu turno de practicar con el siguiente ejercicio:**

#### Práctica:

Realiza un programa en el cual se le pregunte al usuario la siguiente información:

1. Nombre
2. Edad
3. Lugar de nacimiento
4. Licenciatura o programa de pregrado que cursa

Toda esa información debe ser guardada en una variable que se llame Perfil.

## Arreglos y vectores

Los vectores pueden ser creados con distintos tipos de datos, por ejemplo, un vector booleano quedaría de esta manera:

```
> z <- c(TRUE, FALSE, FALSE, TRUE, TRUE)
> z
[1] TRUE FALSE FALSE TRUE TRUE
```

Un vector de strings:

```
> y<- c("Sonia", "Pedro", "Naomi", "Marco")
> y
[1] "Sonia" "Pedro" "Naomi" "Marco"
```

Para extraer valores de los vectores basta con que utilices el índice(s) de los valores deseados:

```
> arreglo<- c(12.5, 32.3, 5.9, 10.2, 8.9, 23.7)
> arreglo
[1] 12.5 32.3 5.9 10.2 8.9 23.7
> arreglo[1]
[1] 12.5
> arreglo[3]
[1] 5.9
> arreglo[5]
[1] 8.9
```

A diferencia de Python y de otros lenguajes de programación, los índices de los arreglos inician desde 1, además es posible extraer varios valores a la vez:

```
> arreglo[c(2, 4, 6)]
[1] 32.3 10.2 23.7
```

Se puede cambiar el valor de un vector de la siguiente forma:

```
> arreglo
[1] 12.5 32.3 5.9 10.2 8.9 23.7
> arreglo[2]<- 100
> arreglo
[1] 12.5 100.0 5.9 10.2 8.9 23.7
```

Una ventaja de lenguaje R es que cuando tienes un vector y deseas extraer valores bajo ciertas condiciones puedes usar de la función `which()` y poner la condición, por ejemplo, si en el vector llamado Arreglo deseas conocer qué elementos están por encima de 10, lo puedes hacer de la siguiente manera:

```
> which(arreglo > 10)
[1] 1 2 4 6
```

Lo cual es correcto, ya que únicamente los elementos 3 y 5 son menores que 10.

Los arreglos o vectores pueden tener nombres en cada uno de sus elementos, para eso observa el siguiente ejemplo:

```
> datos<- c(10, 20, 30)
> datos
[1] 10 20 30
> names(datos)<- c("Primero", "Segundo", "Tercero")
> datos
Primero Segundo Tercero
      10       20       30
```

Los vectores son agrupaciones que deben contar con los mismos tipos de datos. En caso de que se desee contar con una variable con distintos tipos de datos se puede realizar mediante las listas:

```
> info<- list(FALSE, 100, 200, "ingresos")
> info
[[1]]
[1] FALSE

[[2]]
[1] 100

[[3]]
[1] 200

[[4]]
[1] "ingresos"
```

## If else

---

Para trabajar con el condicional if-else, se tiene la siguiente sintaxis:

```
if(condición)
{
    Se ejecuta el código entre las llaves
}
else
{
    Se ejecuta el código entre las llaves
}
```

Por ejemplo, si quieres generar un número aleatorio entre 0 y 100 en el programa asignarás a la variable valor Alto (en caso de que el número sea mayor que 50) y Bajo si es menor que este:

```
> dato <- runif(1, 0, 100)
> if(dato > 50){ valor="alto"} else{valor="bajo"}
> dato
[1] 88.18495
> valor
[1] "alto"
```

Prueba para otros valores generados:

```
> dato <- runif(1, 0, 100)
> if(dato > 50){ valor="alto"} else{valor="bajo"}
> dato
[1] 66.71788
> valor
[1] "alto"
> dato <- runif(1, 0, 100)
> if(dato > 50){ valor="alto"} else{valor="bajo"}
> dato
[1] 30.752
> valor
[1] "bajo"
```

## Ciclos

---

La sintaxis del ciclo *for* es la siguiente:

```
for (variable in rango)
{
}
```

Por ejemplo, utiliza un ciclo *for* para que genere 15 iteraciones para la variable *x* (empezando en 1 hasta llegar al 15):

```
> for(x in 1:15)
+ print(x)
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
```

Ahora genera un vector de datos y haz que este se itere en una secuencia del ciclo *for*:

```
> nombres<- c("Laura", "Teresa", "Erick", "Mario", "Aurora")
> for(i in seq_along(nombres))
+ print(nombres[i])
[1] "Laura"
[1] "Teresa"
[1] "Erick"
[1] "Mario"
[1] "Aurora"
```

### Ahora es tu turno para practicar:

#### Ejercicio:

En este ejercicio se pondrá en práctica todo lo aprendido en este bloque.

Realiza un programa en donde se le pregunte al usuario por un número entero que sea mayor o igual a 1 y menor que 10. Una vez que el usuario introduzca el número, el programa debe ser capaz de calcular el número factorial del mismo.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECNILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECNILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECNILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educativo y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.