



ACTIVIDAD 1

Tipo actividad: Preparación de un entorno en R

"R" es un lenguaje de programación y un entorno de desarrollo utilizado principalmente para análisis estadístico, computación numérica y gráficos. Es un software de código abierto y es ampliamente utilizado en estadística.

R proporciona una amplia gama de funciones estadísticas y gráficas listas para usar, y también permite a los usuarios crear sus propias funciones y paquetes para realizar análisis personalizados. Además, su comunidad activa contribuye con una gran cantidad de paquetes adicionales que extienden su funcionalidad. Similar a como sucede con Python. R también incluye interoperabilidad con Python y otros lenguajes de programación.

Para instalar R ve al sitio web oficial de R (<https://cran.r-project.org/>) y elige el enlace de descarga correspondiente a tu sistema operativo (Windows, macOS o Linux). Posteriormente se debe seleccionar el vínculo "install R for the first time" y luego en el enlace Download R for Windows, como se ve en la figura 1.



The screenshot shows the official CRAN website for downloading R-4.3.3 for Windows. The page includes a sidebar with navigation links such as 'CRAN', 'Mirrors', 'What's new?', 'Search', 'CRAN Team', 'About R', 'R Homepage', 'The R Journal', 'Software', 'R Sources', 'R Binaries', 'Packages', 'Task Views', 'Other', 'Documentation', 'Manuals', 'FAQs', 'Contributed', 'Donations', and 'Donate'. The main content area features a large download button for 'R-4.3.3 for Windows' (79 megabytes, 64 bit), a link to the 'README on the Windows binary distribution', and a link to 'New features in this version'. Below this, there is a note about UCRT requirements and a frequently asked questions section. The page also includes links to 'Other builds' and a note about webmasters.

Figura 1: Página de descarga de R.

Una vez que se haya descargado el instalador, se debe ejecutar y seguir las instrucciones del instalador, dejando todas las opciones por defecto.

Así como sucede con Python, R también tiene algunos entornos de desarrollo integrados (IDE). Oficialmente R ofrece el IDE R-studio que tiene muchas herramientas para trabajar con R, limpiar el código, analizar sucesos, instalar bibliotecas, entre otras tareas asociadas al desarrollo con R.

Instalación de R studio

Para instalar R studio se debe ir a la página de descarga oficial en: <https://posit.co/download/rstudio-desktop/> haciendo click en el botón DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS.

Al descargar el archivo, se debe ejecutar y seguir el paso a paso del instalador dejando las rutas y las configuraciones por defecto.

Al finalizar la instalación se debe ejecutar el programa RStudio instalado (el ícono se ve como en la figura 2)

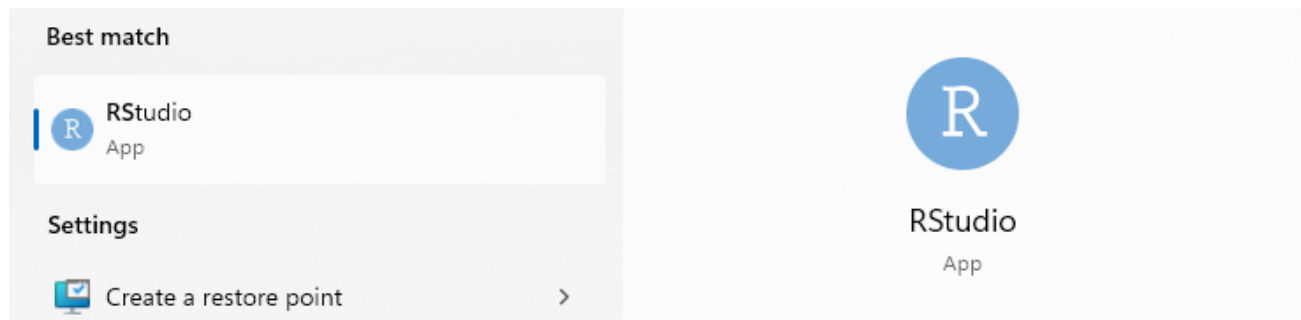


Figura 2 ejecutar RStudio.

Al abrir por primera vez, se pedirá establecer la ruta en donde está instalado R (figura 3) dejaremos todo por defecto y haremos clic en el botón ok.

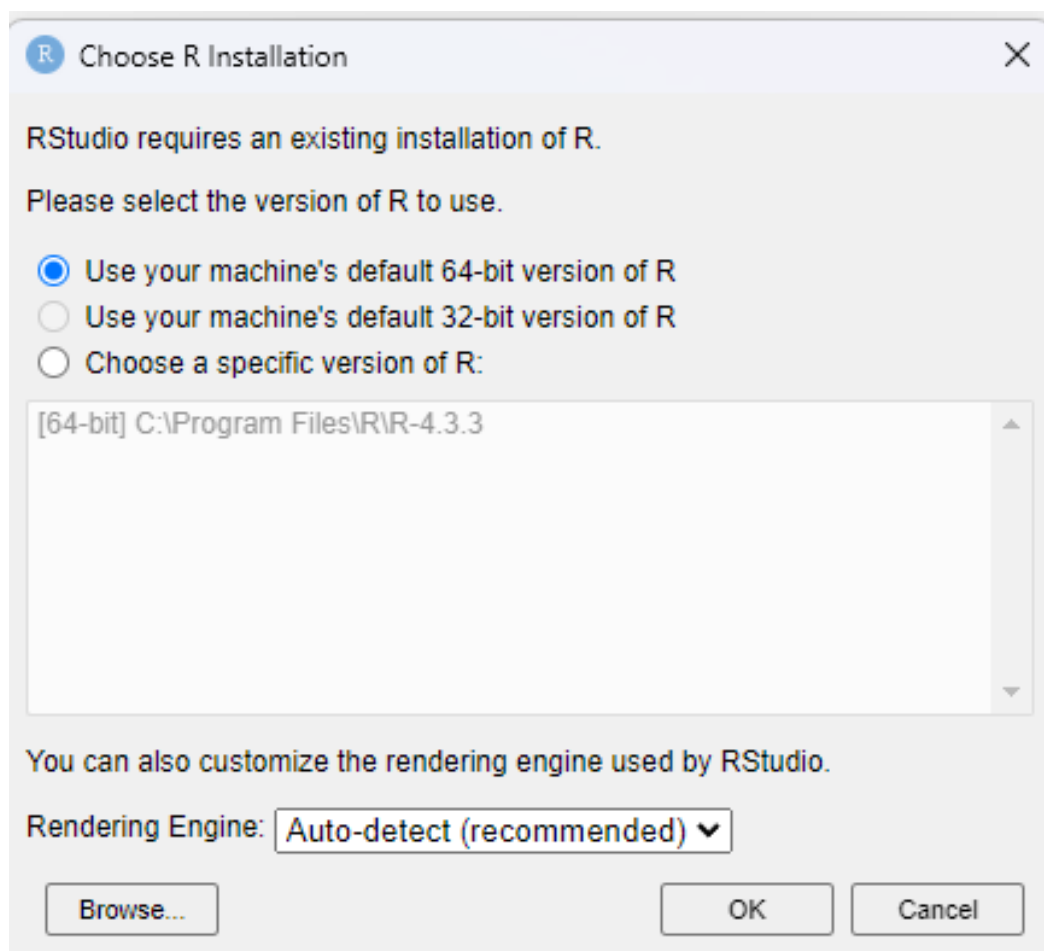


Figura 3 ejecución de R studio por primera vez.



Luego se verá la imagen de la figura 4 en donde podemos empezar a programar

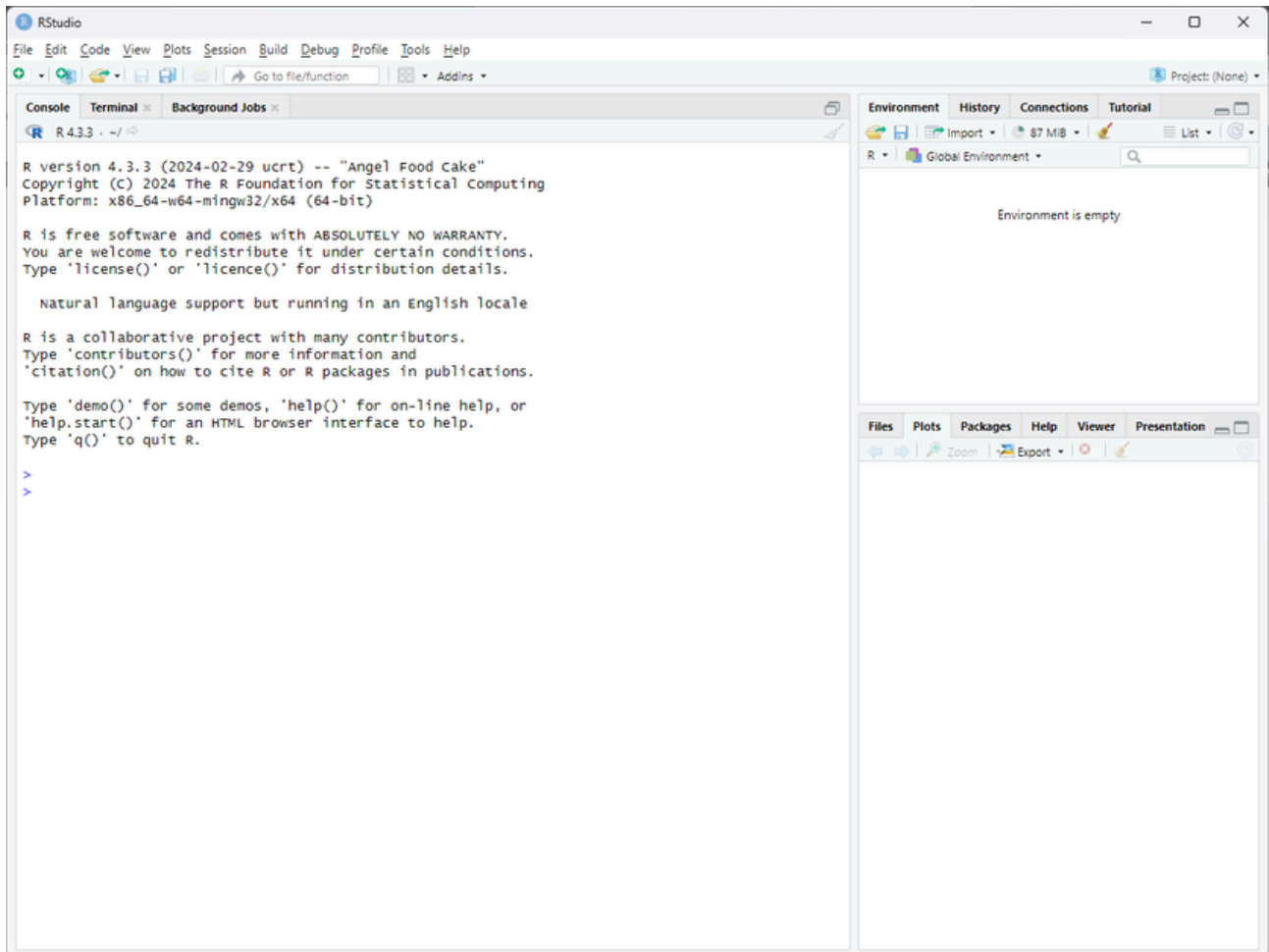


Figura 4: Entorno R Studio.

Sintaxis de R

R es un lenguaje orientado a objetos, e interpretado (se asemeja en estas características a Python) Esto significa que si escribimos comandos en el panel con el nombre “console” en donde se ven los símbolos “>” estos se ejecutarán inmediatamente y se apreciará el resultado.



Se pueden escribir operaciones matemáticas, y R las ejecutará de inmediato, por ejemplo $5+5$, $8/4$, entre otras (figura 5).

```
>  
> 5+5  
[1] 10  
> 8/4  
[1] 2  
> 4/8  
[1] 0.5  
> |
```

Figura 5: Aritmética en R.

Para crear variables se debe asignar un valor al nombre de la variable. A diferencia de otros lenguajes, en R no se asigna con el operador $=$ sino con el operador $\leftarrow$

Por ejemplo, para crear variables se puede escribir:

```
> dato1 <- 500
```

```
> dato2 <- 22
```

```
> dato1 / dato2
```

```
[1] 22.72727
```

Con lo que se asigna el valor 500 a la variable “dato1”, el valor 22 a la variable “dato2” y luego se divide el contenido de la variable “dato1” entre el contenido de la variable “dato2”.

En el panel de la derecha se puede observar las variables creadas y su contenido (figura 6)

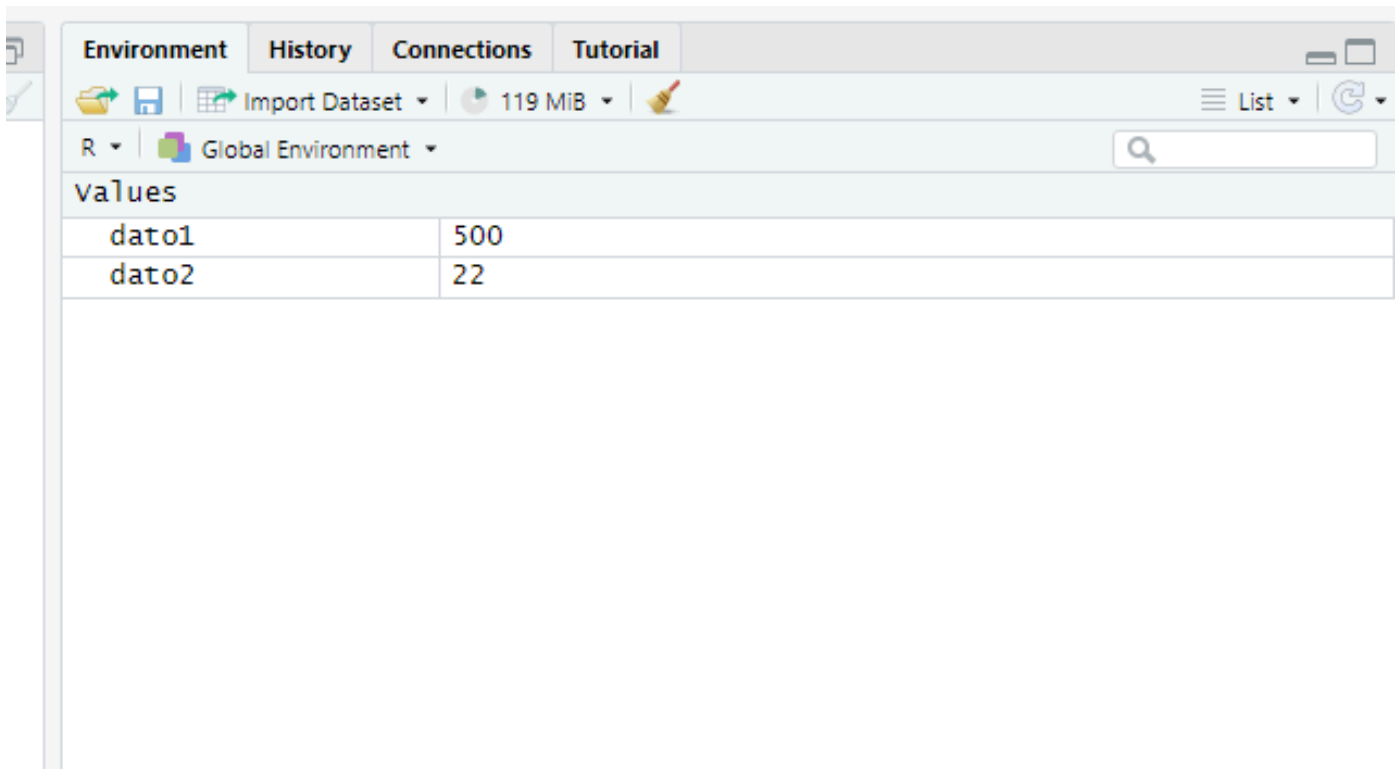


Figura 6: Panel de entorno.

En R los comentarios se escriben igual que en Python, empleando el símbolo '#'.

Para mostrar datos en pantalla se cuenta con la función `print`, por ejemplo `print(dato1)` mostrará el contenido (500) en pantalla. Algunas funciones básicas que se pueden usar sin incluir ninguna librería seon: `sum()`, `mean()`, `sd()` y `length()`

Tipos de datos

R incluye varios tipos de datos para el manejo de la información como son:

- Tipos numéricos (números flotantes, enteros y complejos)

- Cadenas de caracteres (strings)

- Datos booleanos



Operaciones aritméticas

Las operaciones aritméticas que admite R son:

Operador	Nombre	ExampleEjemplo
+	Suma	$x + y$
-	Resta	$x - y$
*	Multiplicación	$x * y$
/	División	x / y
^	Potenciación	$x ^ y$
%%	Módulo (residuo de la división)	$x \% \% y$
%/%	Division entera (la salida es un entero)	$x \% \% y$



Operaciones de comparación

Operador	Nombre	Ejemplo
==	Es igual a	$X == y$
!=	Es diferente de	$X != y$
>	Mayor que	$X > y$
<	Menor que	$X < y$
>=	Mayor o igual que	$X >= y$
<=	Menor o igual que	$X <= y$



Operadores lógicos

Operador	Nombre
&	AND de elemento a elemento
&&	And lógica
	OR de elemento a elemento
	OR lógico
!	NOT lógico

Operadores propios de R

Operador	Descripción	Ejemplo
:	Crea una serie de números en una secuencia	X <- 1:10 (asigna los números de 1 a 10 como array)
%in%	Encuentra si un elemento pertenece a un vector	X %in% y
%*%	Multiplicación de matrices	X<- matriz1 %*% matriz2



Estructuras de decisión y de control

Como en todo lenguaje, se tienen condicionales if – else y ciclos for y while.

Algunos ejemplos:

```
# Ejemplo de condicionales IF-ELSE
```

```
a <- 33b <- 33if (b > a) { print("b is greater than a")} else if (a == b)
{ print ("a and b are equal")}
```

```
# Ejemplo de ciclos While
```

```
i <- 1while (i < 6) { print(i) i <- i + 1}
```

```
# Ejemplo de ciclos for
```

```
for (x in 1:10) { print(x)}
```

```
# ejemplo de creación de una función
```

```
my_function <- function() { # Crea una function llamada function print("Hello
World!")}
```

Ejercicio: Utilizando R y R Studio crear los siguientes programas:

1. Programa que, dados 3 números encuentre cual es el mayor de los tres.
2. Programa que permita imprimir en pantalla los números impares
3. Programa que permita encontrar los primeros 10 números primos.
4. Programa que permita conocer los divisores de un número.