

Actividad | 1 | Análisis de Conceptos

Métodos Numéricos

Ingeniería en Desarrollo de Software



TUTOR: Miguel Angel Rodríguez Vega

ALUMNO: Karol Ochoa Beltran

FECHA: 06 de noviembre 2025

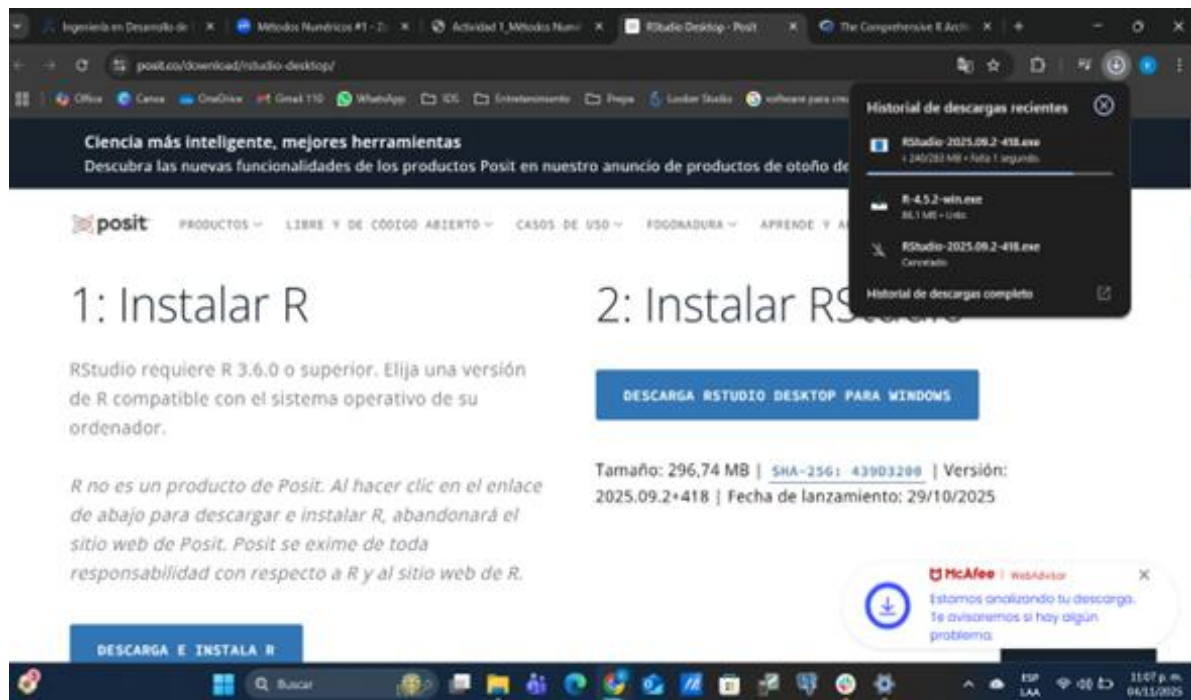
Índice

Desarrollo	2
Descarga de Rstudio.....	2
<i>Instalación R.....</i>	<i>2</i>
<i>Instalación RStudio</i>	<i>4</i>
Carga de Valores_numericos.R.....	6
Ejecución de Valores_numericos.R	6
Referencias.....	10

Desarrollo

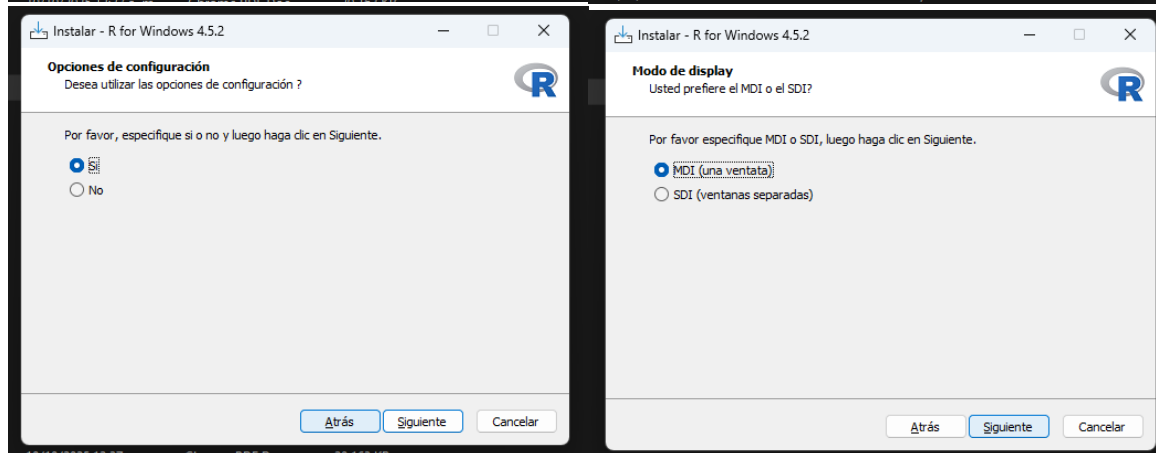
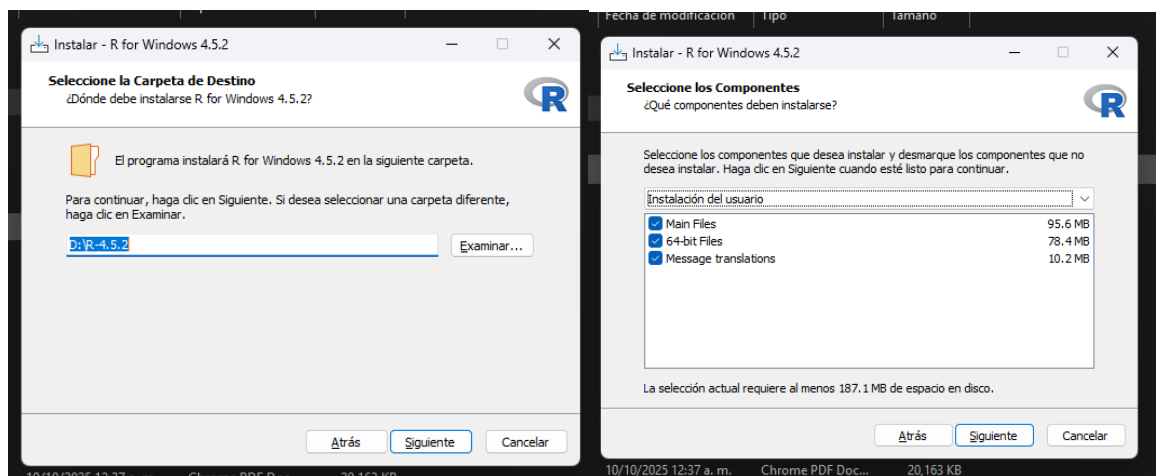
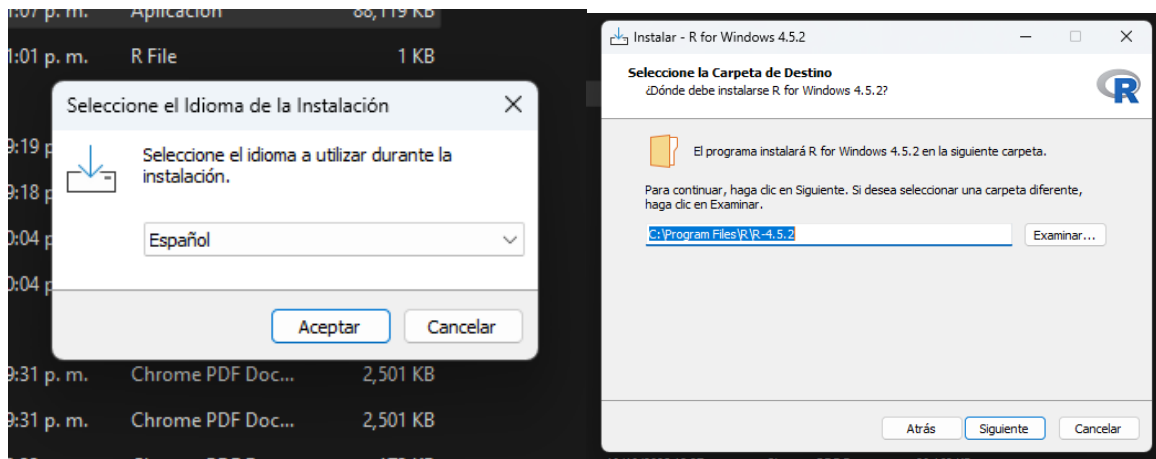
Descarga de Rstudio

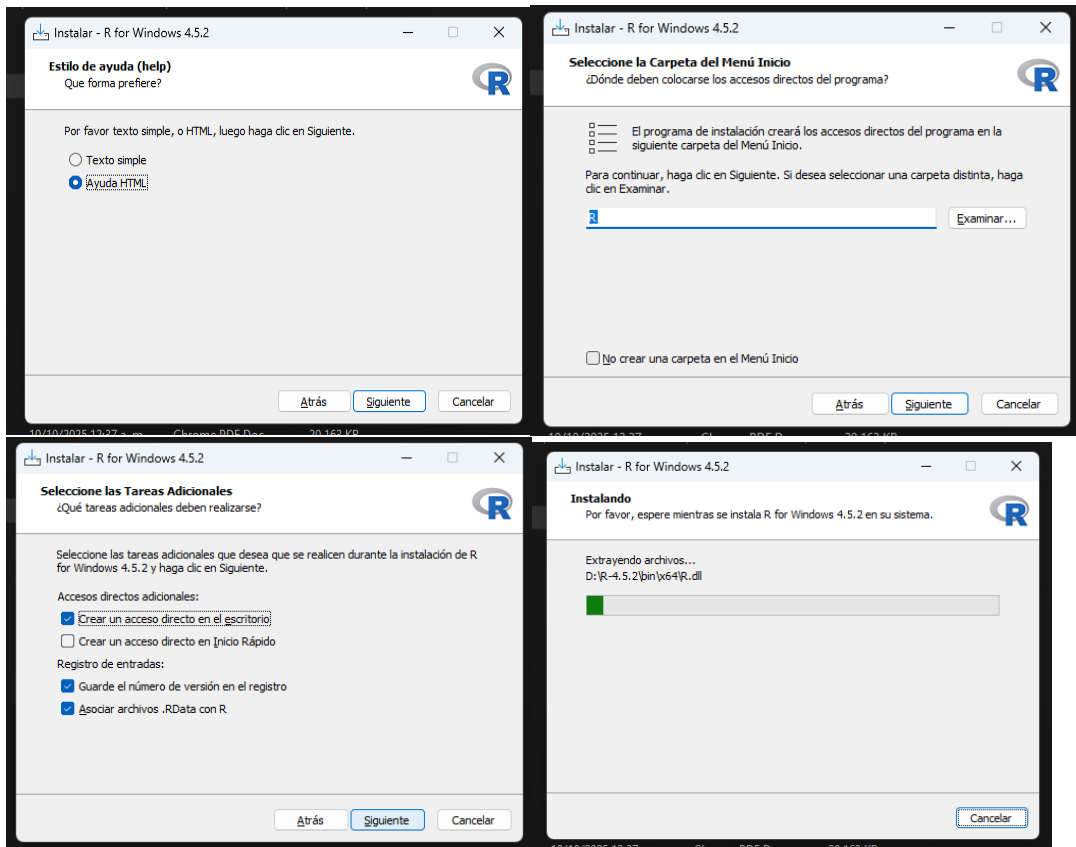
Primero descargaremos las aplicaciones de la página que se nos proporcionó en el pdf donde vienen las instrucciones para realizar la actividad. Es muy importante instalar tanto R como RStudio para poder ejecutar el lenguaje de programación.



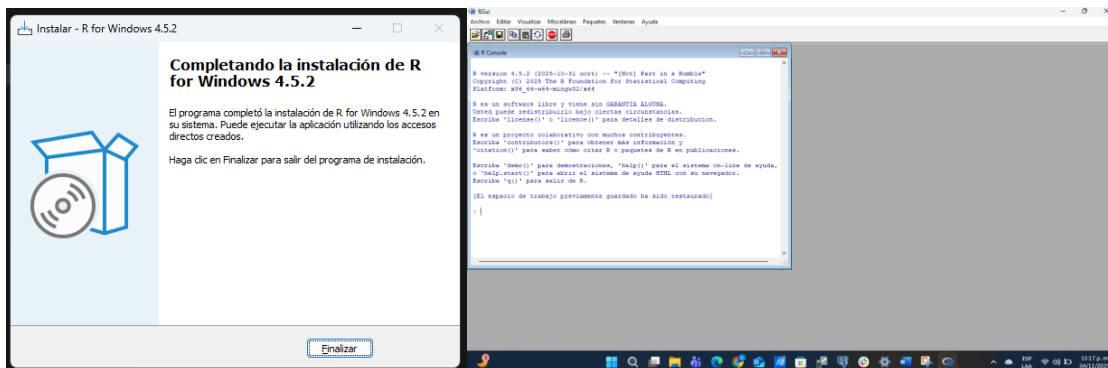
Instalación R

R será el motor que nos permitirá ejecutar el código en RStudio, se seleccionarán las siguientes configuraciones:





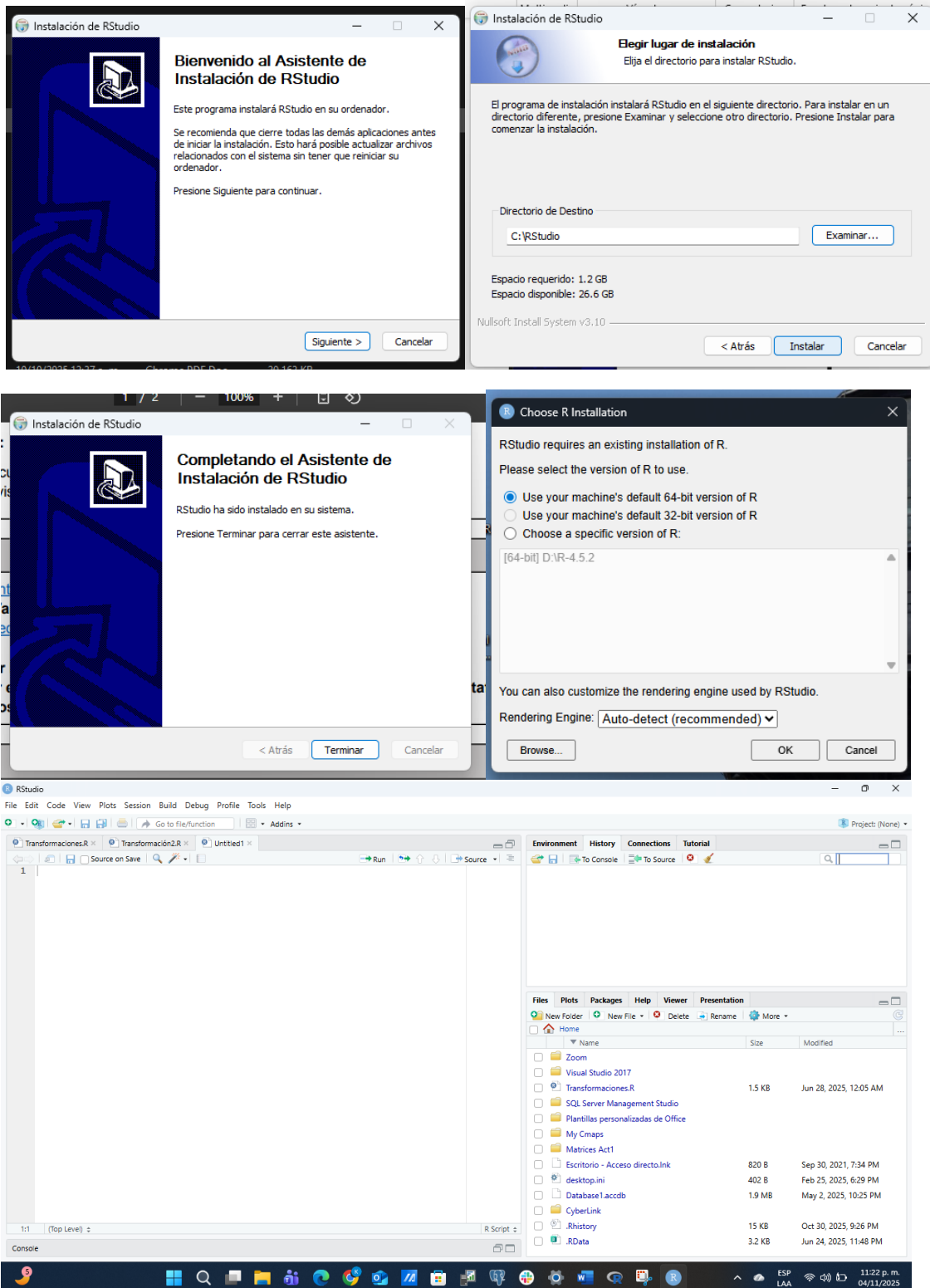
Llegados a este punto ya hemos finalizado con la instalación de la app.



Instalación RStudio

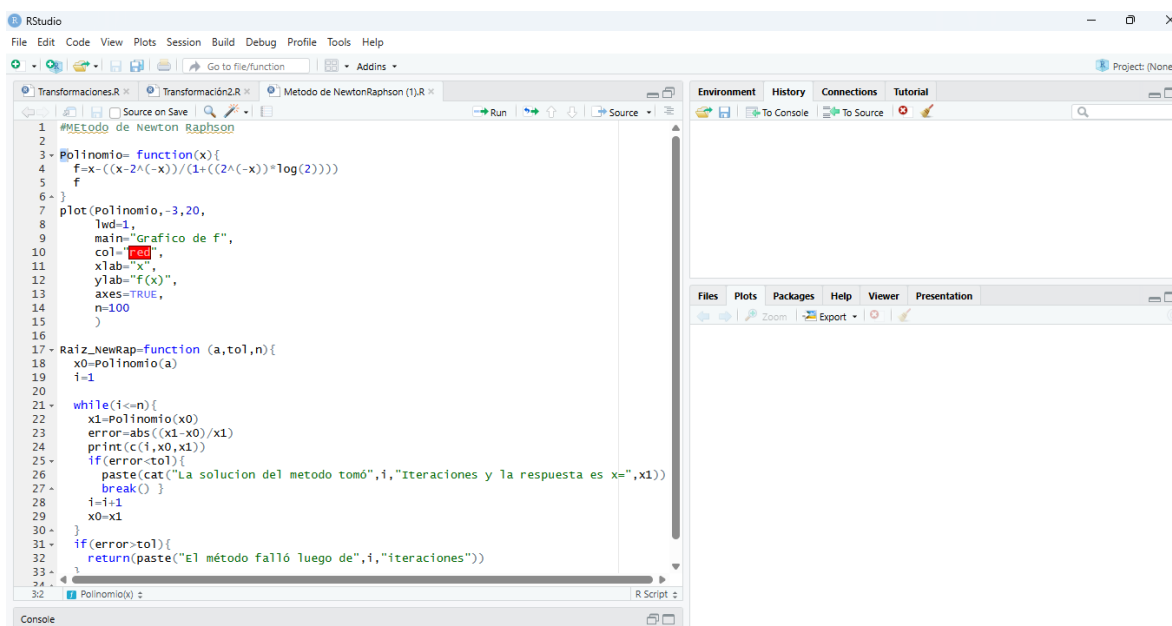
RStudio es el entorno que nos permitirá interactuar con el lenguaje R pero de una manera mucho más sencilla e intuitiva. En el caso de esta aplicación se seleccionan las

configuraciones predeterminadas.



Carga de Valores_numericos.R

En la imagen podemos observar diferentes operaciones y comandos utilizados en el lenguaje de programación R, los cuales permiten realizar diferentes operaciones matemáticas.



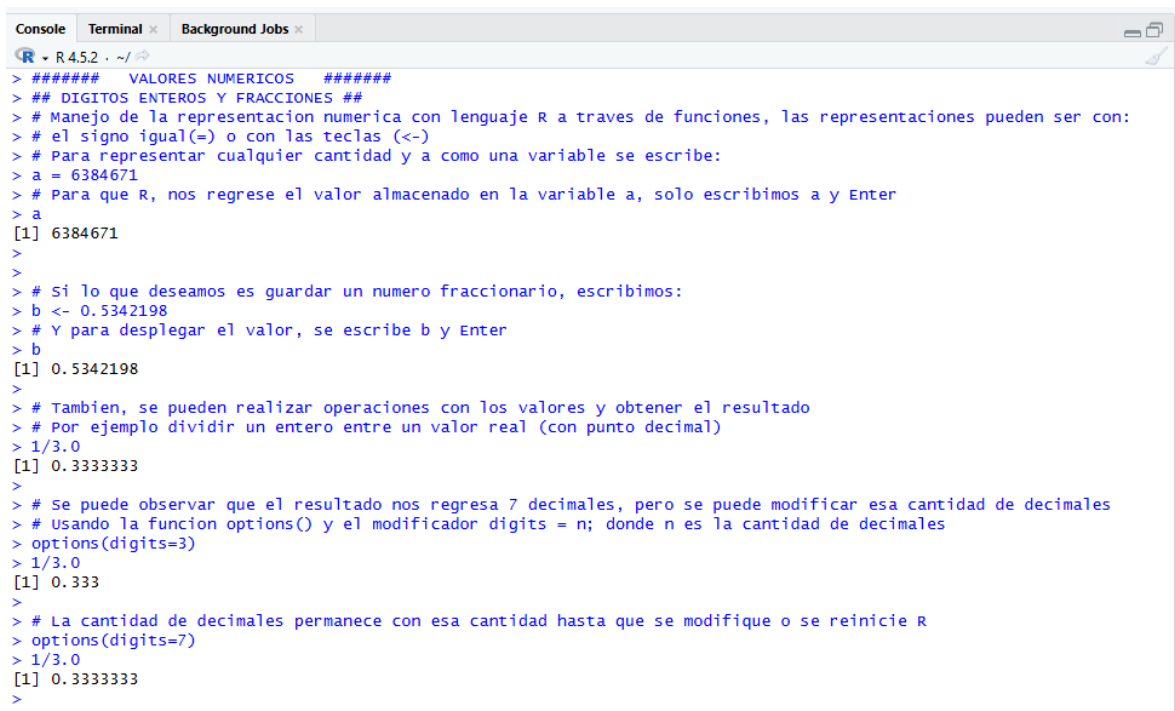
Ejecución de Valores_numericos.R

En la primera imagen tenemos el apartado de “Valores numéricos”, en este trabajamos con 5 ejercicios, que son los siguientes:

1. Asignamos un valor a la variable “a”, esto se hace con el símbolo “=” y al escribir el nombre de la variable y dar enter se regresa el valor que se le asignó.
2. Usando “<=” al igual que el ejemplo anterior sirve para asignarle valor a una variable, siendo este es el modo más común para hacer esta tarea.
3. En este ejemplo se define que se pueden hacer operaciones matemáticas con diferentes símbolos, en este caso en particular se usa la diagonal para dividir, aunque también se puede multiplicar con el asterisco o con los paréntesis,

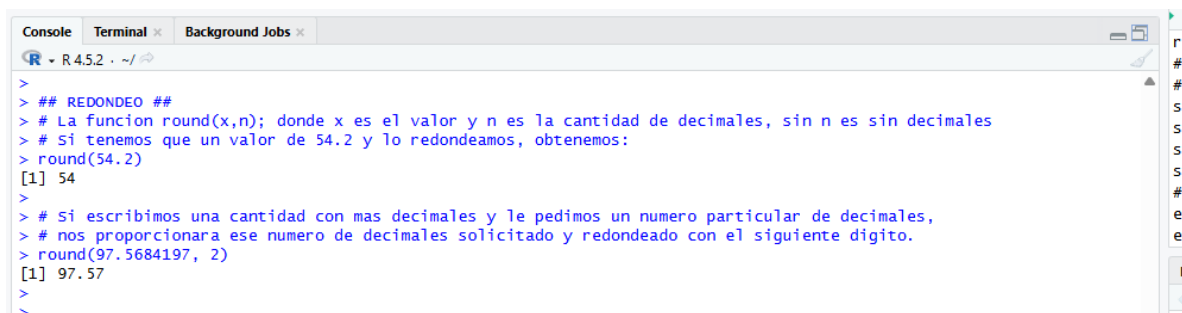
hacer sumas y restas con sus respectivos símbolos, etc.

4. Retomando el ejemplo anterior, visualizamos que el resultado fue de 7 decimales, en RStudio podemos establecer cuantos decimales queremos ver al escribir “options(digits=3)”, siendo el comando “digits” el que define el número de decimales.
5. Continuando con el inciso anterior nos indica que al modificar el numero de decimales mostrados, se seguirá mostrando así hasta que nosotros indiquemos una instrucción diferente.



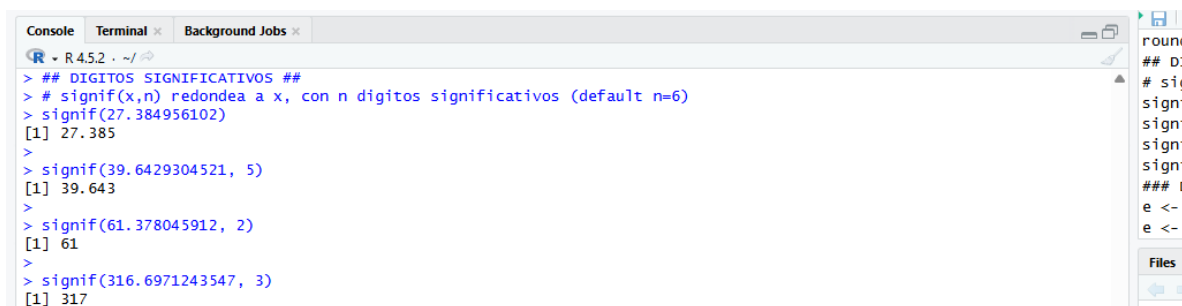
```
> ##### VALORES NUMERICOS #####
> ## DIGITOS ENTEROS Y FRACCIONES ##
> # Manejo de la representacion numerica con lenguaje R a traves de funciones, las representaciones pueden ser con:
> # el signo igual(=) o con las teclas (<-)
> # Para representar cualquier cantidad y a como una variable se escribe:
> a = 6384671
> # Para que R, nos regrese el valor almacenado en la variable a, solo escribimos a y Enter
> a
[1] 6384671
>
> # Si lo que deseamos es guardar un numero fraccionario, escribimos:
> b <- 0.5342198
> # Y para desplegar el valor, se escribe b y Enter
> b
[1] 0.5342198
>
> # Tambien, se pueden realizar operaciones con los valores y obtener el resultado
> # Por ejemplo dividir un entero entre un valor real (con punto decimal)
> 1/3.0
[1] 0.3333333
>
> # Se puede observar que el resultado nos regresa 7 decimales, pero se puede modificar esa cantidad de decimales
> # Usando la funcion options() y el modificador digits = n; donde n es la cantidad de decimales
> options(digits=3)
> 1/3.0
[1] 0.333
>
> # La cantidad de decimales permanece con esa cantidad hasta que se modifique o se reinicie R
> options(digits=7)
> 1/3.0
[1] 0.3333333
>
```

En “Redondeo” la imagen nos muestra la función “round” la cual nos permite redondear un número, por si sola redondea a un numero entero, pero al paréntesis le podemos agregar una coma y el número de decimales que queremos que nos muestre para tener mayor control de los datos que queremos visualizar.



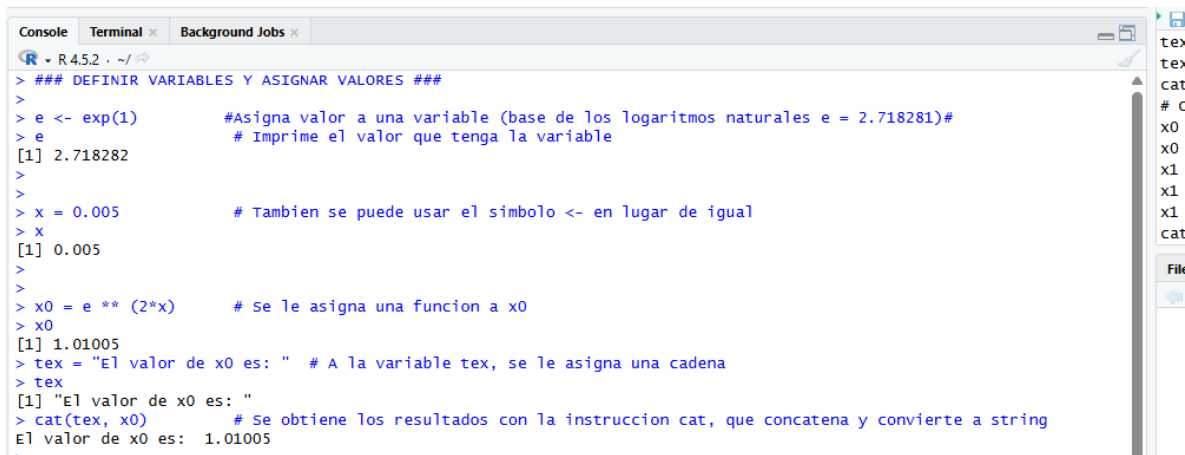
```
R - R 4.5.2 - ~/
>
> ## REDONDEO ##
> # La función round(x,n); donde x es el valor y n es la cantidad de decimales, sin n es sin decimales
> # Si tenemos que un valor de 54.2 y lo redondeamos, obtenemos:
> round(54.2)
[1] 54
>
> # Si escribimos una cantidad con mas decimales y le pedimos un numero particular de decimales,
> # nos proporcionara ese numero de decimales solicitado y redondeado con el siguiente digito.
> round(97.5684197, 2)
[1] 97.57
>
>
```

A continuación, tenemos “Dígitos significativos”. En ese caso la función delimitará cuales son los dígitos que aportan mayor precisión a la cantidad que se quiere redondear, retomando la dinámica de la coma y el numero para controlar de manera manual la cantidad de dígitos que se mostrarán.



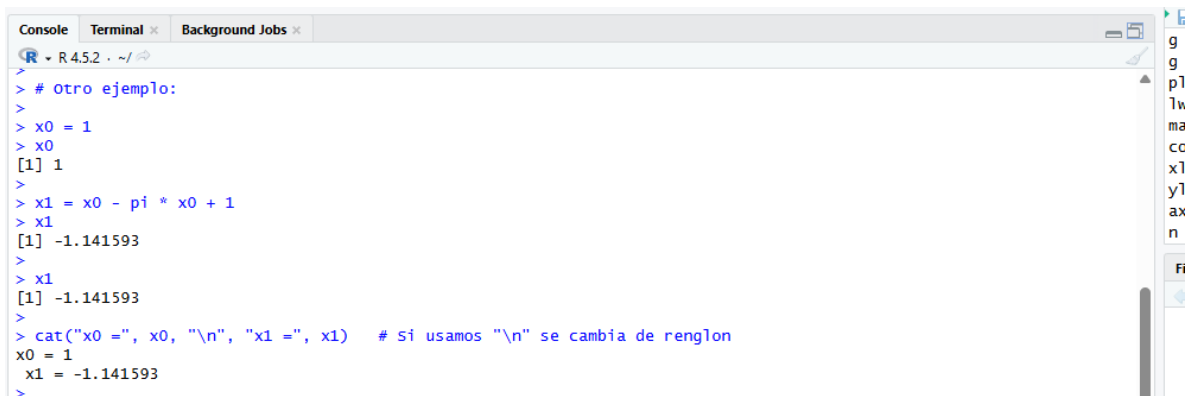
```
R - R 4.5.2 - ~/
> ## DIGITOS SIGNIFICATIVOS ##
> # signif(x,n) redondea a x, con n digitos significativos (default n=6)
> signif(27.384956102)
[1] 27.385
>
> signif(39.6429304521, 5)
[1] 39.643
>
> signif(61.378045912, 2)
[1] 61
>
> signif(316.6971243547, 3)
[1] 317
```

En “Definir variables y asignar valores” retomamos el primer apartado, ya que utilizamos “=” y “<-”, la constante “e” es la base del sistema de logaritmos naturales y tiene un valor aproximado al mostrado en la imagen, x0 representa una variable, cuyo valor es el resultado de la variable “e” multiplicada por 2 que se multiplica por el valor de la variable “x”.



```
Console Terminal Background Jobs
R 4.5.2 ~
> ### DEFINIR VARIABLES Y ASIGNAR VALORES ###
>
> e <- exp(1)      #Asigna valor a una variable (base de los logaritmos naturales e = 2.718281)#
> e               # Imprime el valor que tenga la variable
[1] 2.718282
>
>
> x = 0.005        # Tambien se puede usar el simbolo <- en lugar de igual
> x
[1] 0.005
>
>
> x0 = e ** (2*x)   # Se le asigna una funcion a x0
> x0
[1] 1.01005
> tex = "El valor de x0 es: " # A la variable tex, se le asigna una cadena
> tex
[1] "El valor de x0 es: "
> cat(tex, x0)      # Se obtiene los resultados con la instruccion cat, que concatena y convierte a string
El valor de x0 es: 1.01005
>
```

En este segundo ejemplo del apartado tenemos la variable x0 y la variable x1 que tiene como valor el resultado de la operación que se muestra. Por otro lado, con la función “cat” imprimimos el texto al poner el texto entre comillas indicamos el texto a imprimir y cuando ponemos la variable tal cual se mostrará el valor de esta, agregando “\n” indicamos un cambio de renglón entre el texto.

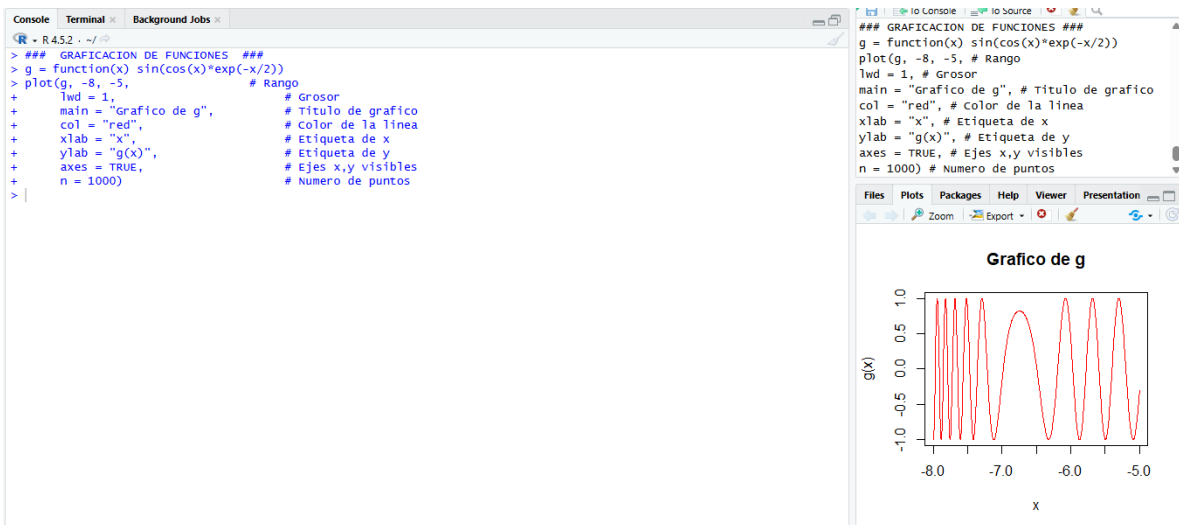


```
Console Terminal Background Jobs
R 4.5.2 ~
> # otro ejemplo:
>
> x0 = 1
> x0
[1] 1
>
> x1 = x0 - pi * x0 + 1
> x1
[1] -1.141593
>
> x1
[1] -1.141593
>
> cat("x0 =", x0, "\n", "x1 =", x1) # Si usamos "\n" se cambia de renglon
x0 = 1
x1 = -1.141593
>
```

En el apartado “Definir funciones y pasar parámetros” podemos observar que se le asigna una función a la variable d, conteniendo en ella una operación matemática y las variables a, b y c, las cuales tienen su valor definido para poder realizar esta operación.

```
Console Terminal Background Jobs
R 4.5.2 ~ /
> ### DEFINIR FUNCIONES Y PASAR PARAMETROS ###
> d = function(a,b,c) b^2-4*a*c # se asigna la funcion a la variable d, con los parametros a,b,c; y la funcion b cu
adrada menos 4 por a por c
> d
function(a,b,c) b^2-4*a*c
> d(2,2,1) # Se llama a la funcion y se le dan los parametros para el calculo y regresa el resultado
[1] -4
```

Para finalizar, la siguiente imagen muestra una manera de generar una grafica “g” es la variable que contiene la función que podemos observar. El comando “plot” define el rango de la gráfica, “ldw” delimita el grosor de la línea, con “main” nombramos el gráfico, “col” define el color que queremos asignarle a la línea, con las funciones “lab” le damos un nombre a cada eje, “axes” indica si se mostrarán los ejes o no y “n” indica el numero de puntos, a mayor cantidad de puntos la línea es más nítida.



Referencias

Ccama, D. A. S. (2022, 4 junio). Aprender Gráficos en R. https://rstudio-pubs-static.s3.amazonaws.com/910622_e69f8d16caef4f72bc51132725d2de35.html#:~:text=Con%20la%20funci%C3%B3n%20plot%20se%20puede%20crear,de%20dispersion%20usand

[o%20la%20funcion%20plot%20.](#)

R: Concatenate and Print. (s. f.).

[https://www.math.ucla.edu/~anderson/rw1001/library/base/html/cat.html#:~:text=cat%20converts%20its%20arguments%20to,fill%20is%20TRUE%20or%20numeric.\)](https://www.math.ucla.edu/~anderson/rw1001/library/base/html/cat.html#:~:text=cat%20converts%20its%20arguments%20to,fill%20is%20TRUE%20or%20numeric.)

SQLPad. (s. f.). Exponents in R: A Comprehensive Guide - Unlock the power of exponents in R programming with this comprehensive guide. Learn through detailed examples how to effectively use exponents in your data a. . . - SQLPad.io.

[https://sqlpad.io/tutorial/exponents-r-comprehensive-guide/#:~:text=R:%20Para%20realizar%20exponenciaciones%20en,cualquier%20tipo%20de%20dato%20num%C3%A9rico.](https://sqlpad.io/tutorial/exponents-r-comprehensive-guide/#:~:text=R:%20Para%20realizar%20exponenciaciones%20en,cualquier%20tipo%20de%20dato%20num%C3%A9rico)

Khan Academy. (s. f.). [https://es.khanacademy.org/v/significant-figures#:~:text=Las%20cifras%20significativas%20son%20el,Creado%20por%20Sal%20Khan.](https://es.khanacademy.org/v/significant-figures#:~:text=Las%20cifras%20significativas%20son%20el,Creado%20por%20Sal%20Khan)

Johnson, R. (2023, 21 noviembre). Logaritmo natural | Reglas, propiedades y ejemplos. Study.com. Recuperado 6 de noviembre de 2025, de <https://study.com/academy/lesson/natural-log-rules-properties-quiz.html>

Link GitHub: