

Actividad 2 - Método de Secante.

Métodos Numéricos

Ingeniería en Desarrollo de Software

Tutor: Miguel Ángel Rodríguez Vega

Alumno: Fernando Pedraza Garate

Fecha: 04 de Noviembre del 2022

Índice

Etapa 1 – Análisis de conceptos.

- Introducción. Pag-3
- Delimitación del problema. Pag-4
- Descarga de R Studio. Pag-5-6
- Carga de valores_numericos.R Pag-7-9
- Ejecución de valores_numericos.R Pag-10-18

Etapa 2 – Método de Secante.

- Carga de Met_Secante.R Pag-19-20
- Ejecución de Met_Secante.R Pag-21-22
- Interpretación de resultados Pag-23
- Conclusión Pag-24
- Referencias Pag-25

Introducción

El análisis formal de conceptos (AFC), en inglés Formal Concept Analysis (FCA), es una teoría matemática y un método para el análisis de datos en cuanto a sus relaciones y estructura. Al aplicarla, la pretensión es que los datos se organicen de manera tal que, sin dejar de responder a la exigencia de rigor de un modelo matemático, se adapten mejor a la forma en que está organizado el pensamiento humano en relación con los conceptos y a su orden. El término fue introducido por Rudolf Wille en 1984, quien se basó en la teoría de retículos y en la teoría matemática del orden desarrollada por Garrett Birkhoff y otros en 1930.

Delimitación del problema.

Los métodos numéricos son aplicaciones de algoritmos mediante las cuales es posible formular y solucionar problemas matemáticos utilizando operaciones aritméticas menos complejas, quiere decir que una vez dominando estas técnicas se puede solucionar cualquier situación, de forma que sea sencillo de resolver, también se conocen como métodos indirectos, quiere decir que un resultado puede tener el mismo valor en distintos sistemas de numeración. Un análisis numérico idealiza y concibe métodos para aprobar, de forma eficiente, las soluciones de problemas expresados matemáticamente. El objetivo principal del análisis numérico es encontrar soluciones aproximadas para problemas complejos.

Descarga de R Studio.



Descarga el IDE de RStudio

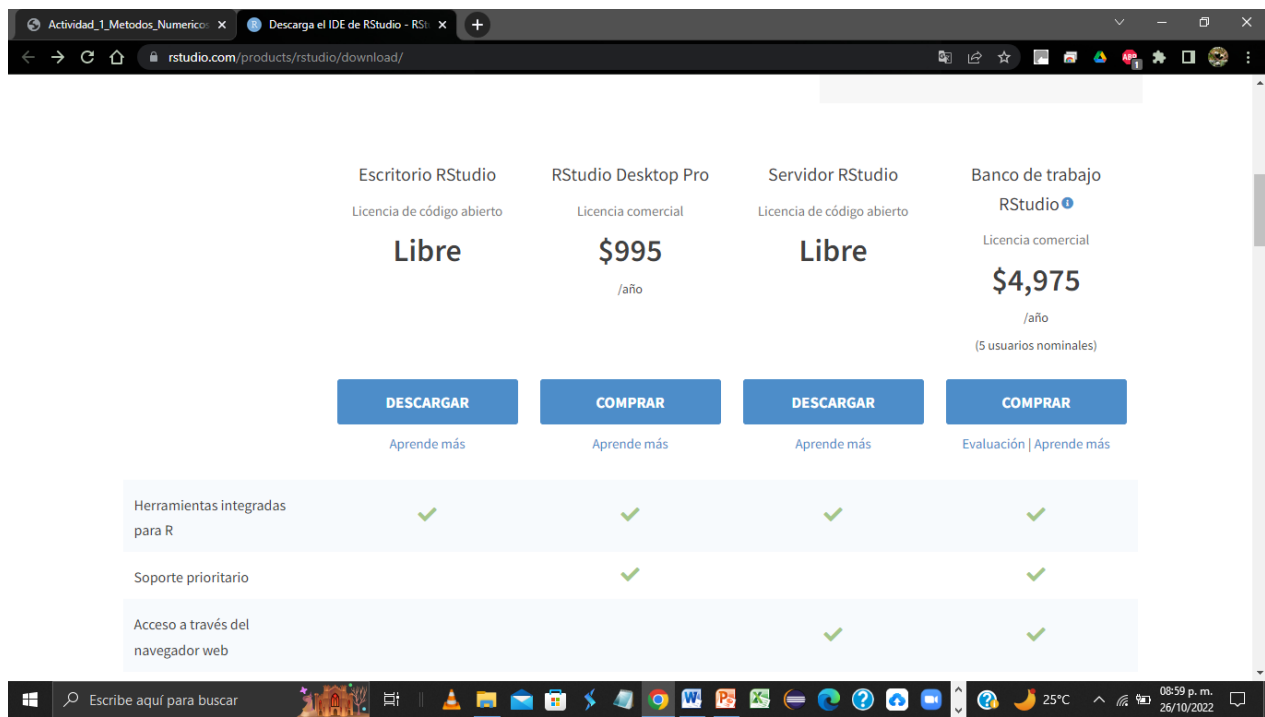
Elige tu versión

RStudio IDE es un conjunto de herramientas integradas diseñadas para ayudarlo a ser más productivo con R y Python. Incluye una consola, un editor de resaltado de sintaxis que admite la ejecución directa de código y una variedad de herramientas sólidas para trazar, ver el historial, depurar y administrar su espacio de trabajo.

RStudio Team

La solución de ciencia de datos profesional recomendada por RStudio para cada equipo. RStudio Team es un paquete del popular software profesional

Ingresamos al sitio web de R Studio.



	Escritorio RStudio	RStudio Desktop Pro	Servidor RStudio	Banco de trabajo RStudio
Licencia de código abierto	Libre	Licencia comercial	Licencia de código abierto	Licencia comercial
	Libre	\$995 /año	Libre	\$4,975 /año
				(5 usuarios nominales)
	DESCARGAR	COMPRAR	DESCARGAR	COMPRAR
	Aprende más	Aprende más	Aprende más	Evaluación Aprende más
Herramientas integradas para R	✓	✓	✓	✓
Soporte prioritario		✓		✓
Acceso a través del navegador web			✓	✓


Seleccionamos la versión libre.

Actividad_1_Metodos_Numericos x Descarga el IDE de RStudio - RSI x +


← → ↻ 🏠 🔒 rstudio.com/products/rstudio/download/#download

Escritorio RStudio 2022.07.2+576 - Notas de la versión [↗](#)

1. instalar r. RStudio requiere R 3.3.0+ [↗](#).
2. Descarga el escritorio de RStudio. Recomendado para su sistema:

 **DESCARGAR RSTUDIO PARA WINDOWS**
2022.07.2+576 | 190,49 MB

Requiere Windows 10/11 (64 bits)



Todos los instaladores

Es posible que los usuarios de Linux necesiten importar la [clave de firma de código pública de RStudio](#) antes de la instalación, dependiendo de la política de seguridad del sistema operativo.

RStudio requiere un sistema operativo de 64 bits. Si está en un sistema de 32 bits, puede usar una [versión anterior de RStudio](#).

sistema operativo	Descargar	Tamaño	SHA-256
-------------------	-----------	--------	---------

Windows taskbar: Escribe aquí para buscar | 25°C | 08:59 p. m. 26/10/2022

Y descargamos desde el enlace según nuestro sistema operativo y ejecutamos el instalador para R Studio.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Environment History Connections Tutorial

R Global Environment

Values

delt	1e-06
error	5.11324316221362e-12
i	9L
n	15
x0	0.472833909000369
x1	0.472833908995256

Functions

df function(x)

Files Plots Packages Help Viewer Presentation

Zoom Export

1:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.2.1 - /

```
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 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.

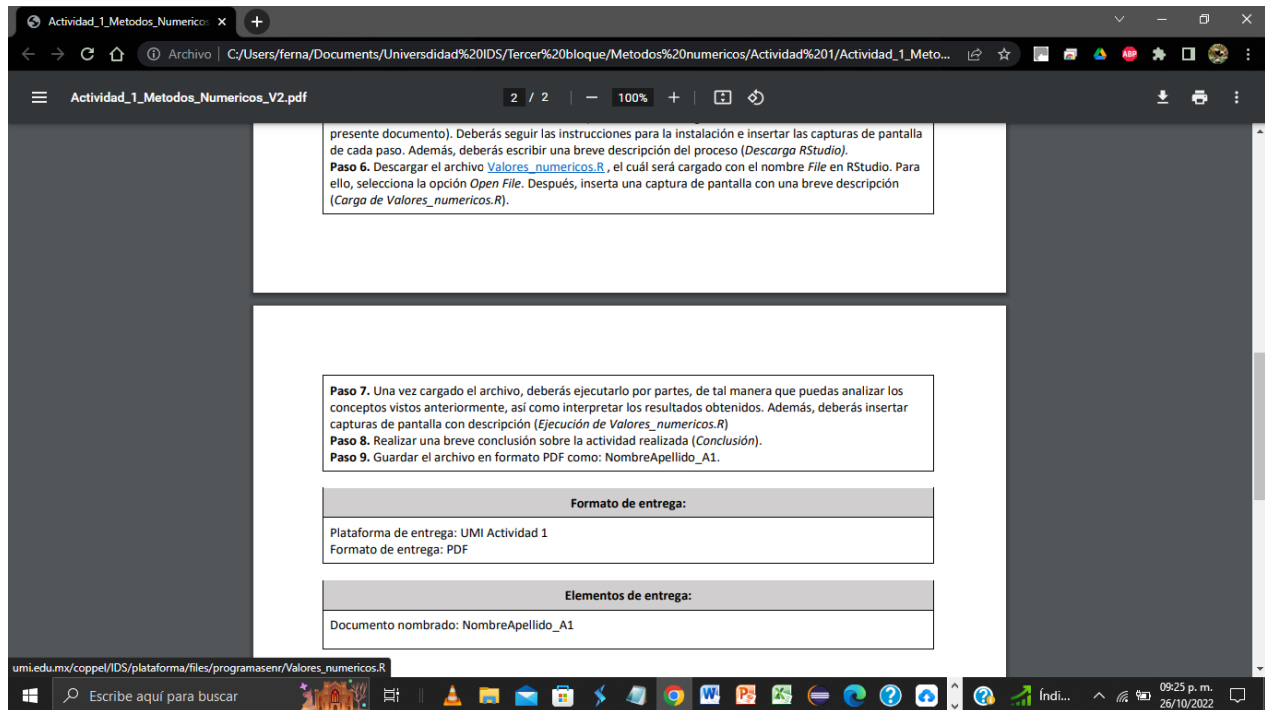
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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

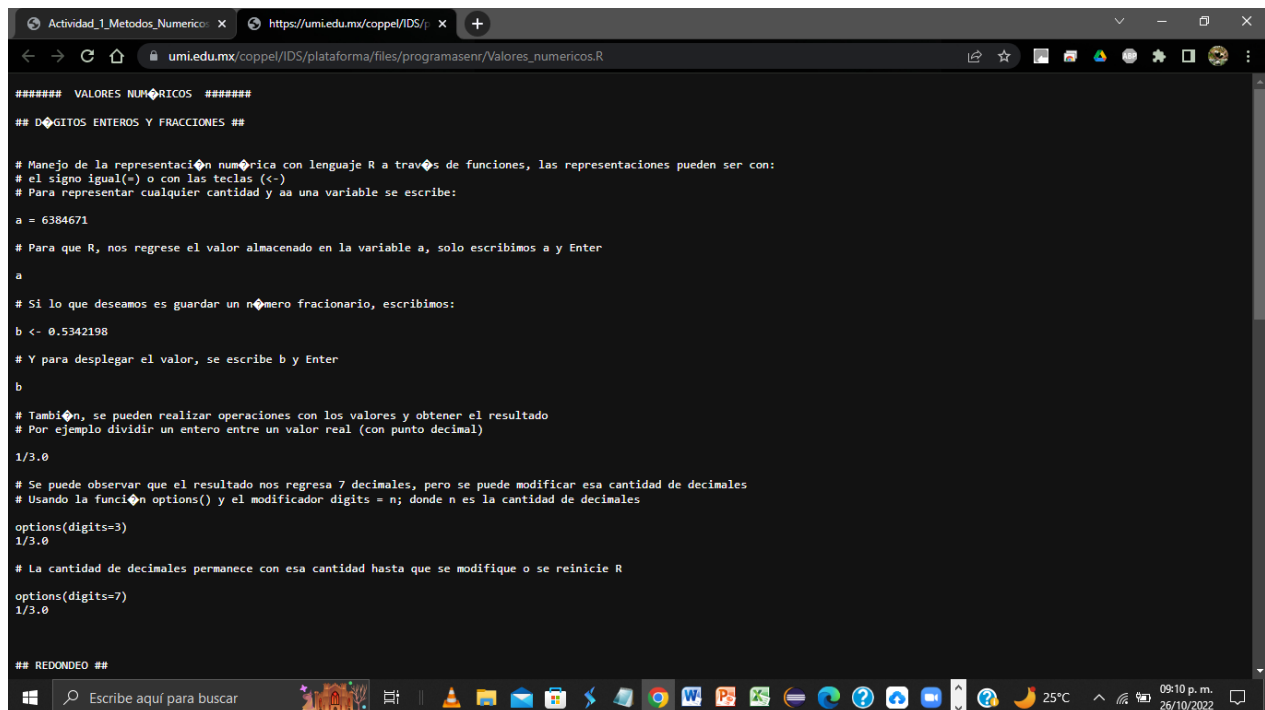
Windows taskbar: Escribe aquí para buscar | 25°C | 08:53 p. m. 26/10/2022

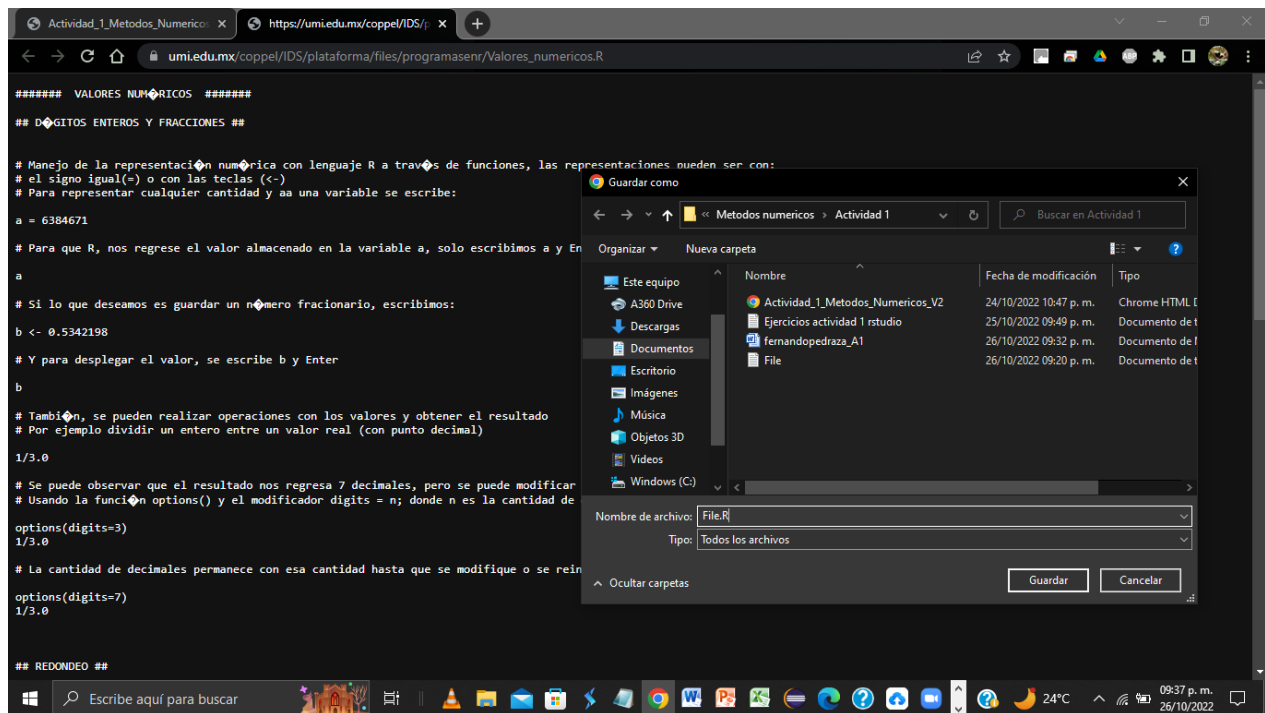
R Studio Instalado.

Carga de valores_numéricos.R

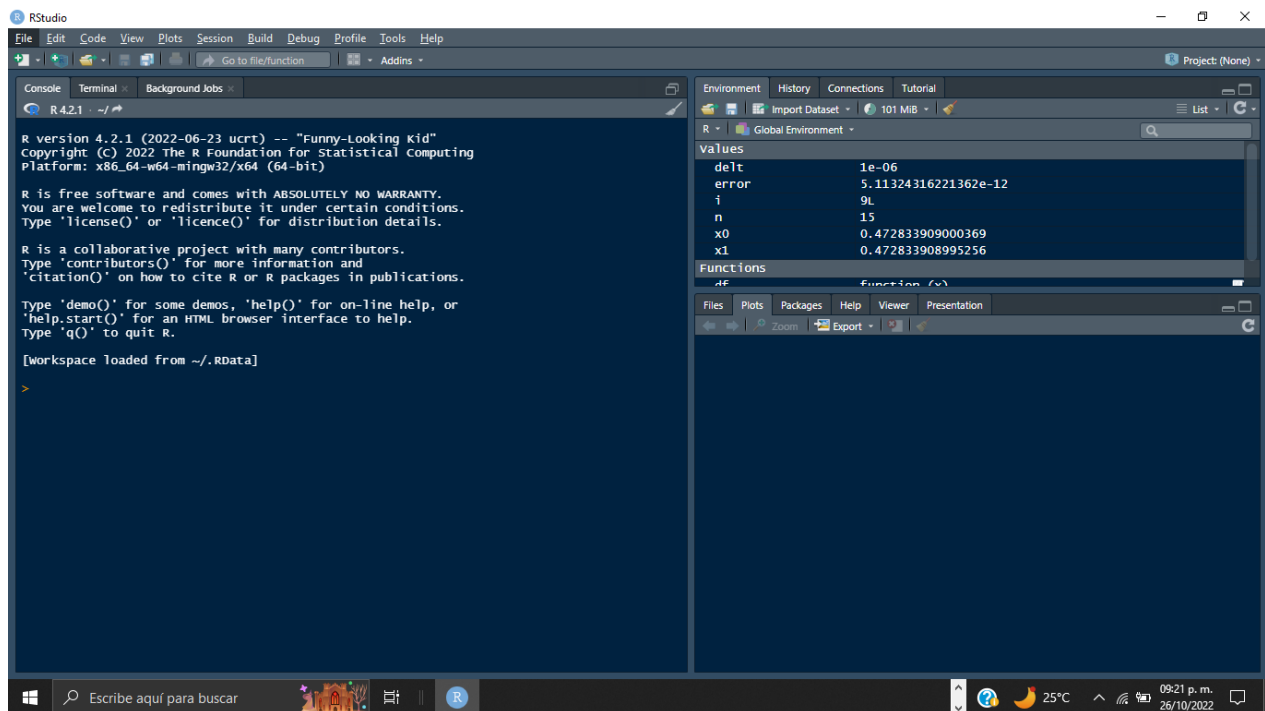


Abrimos el enlace del archivo de valores_numéricos.R

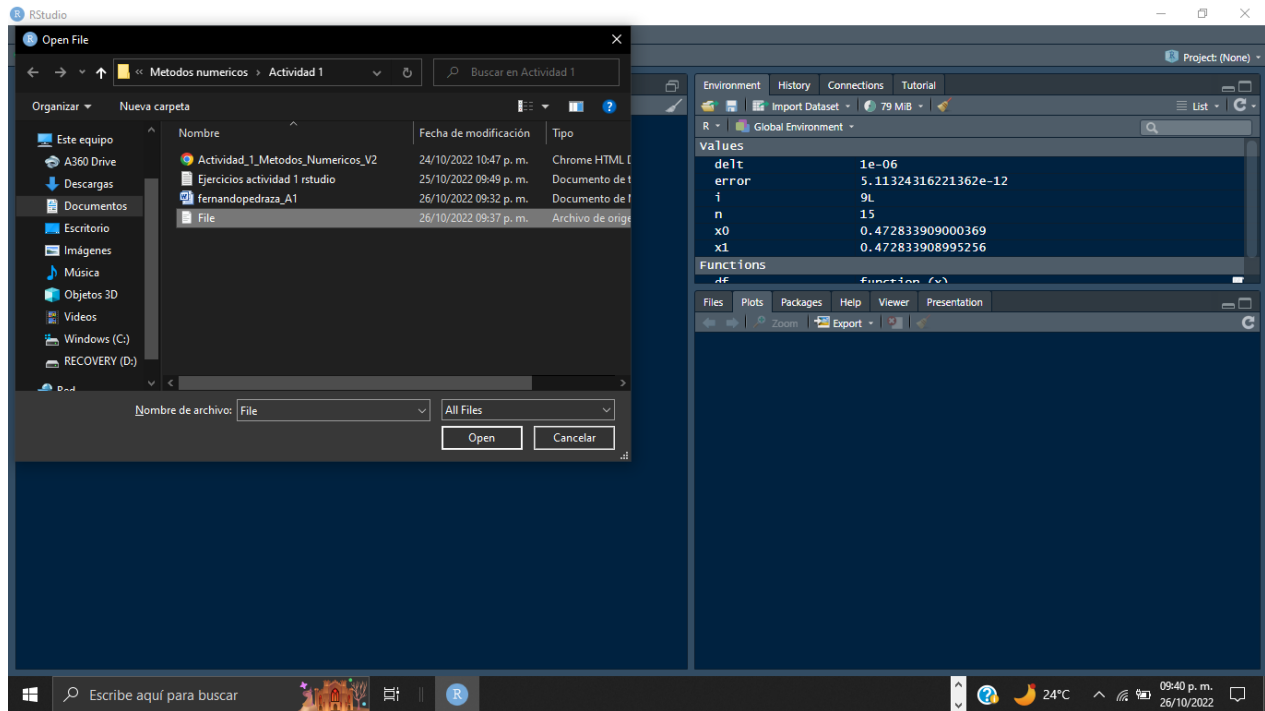




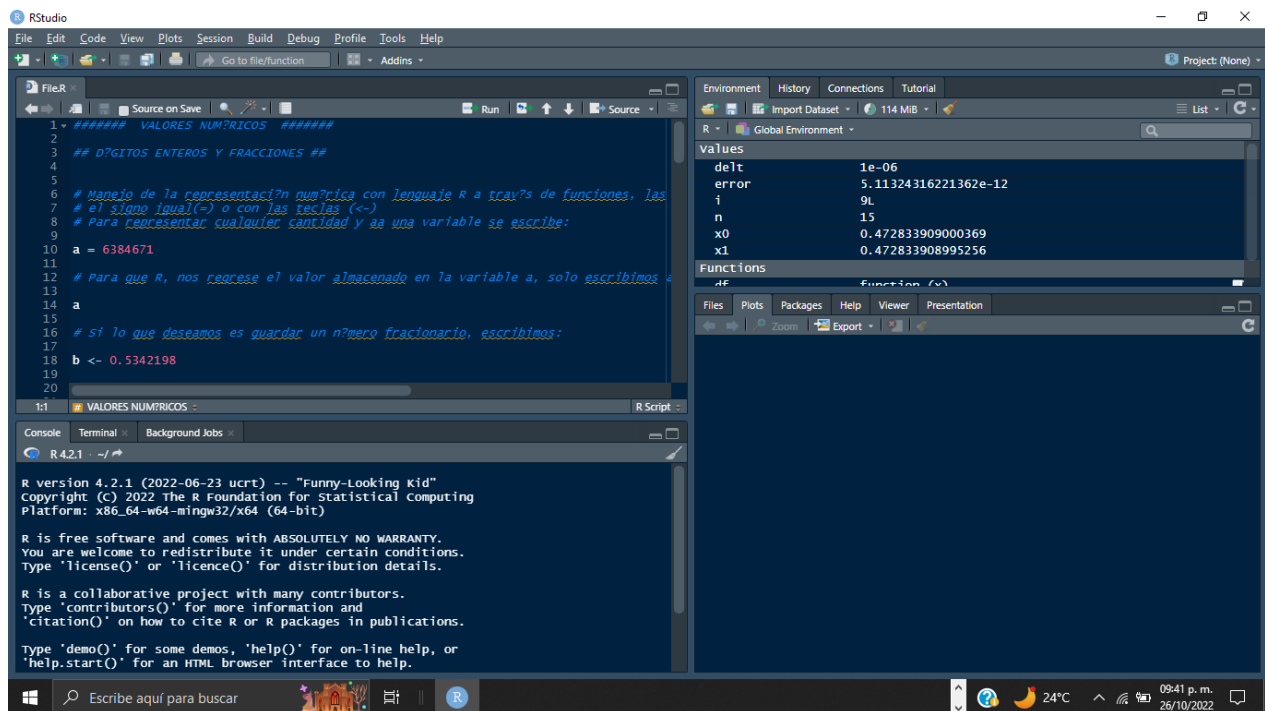
Guardamos el archivo como File.R



Abrimos R Studio

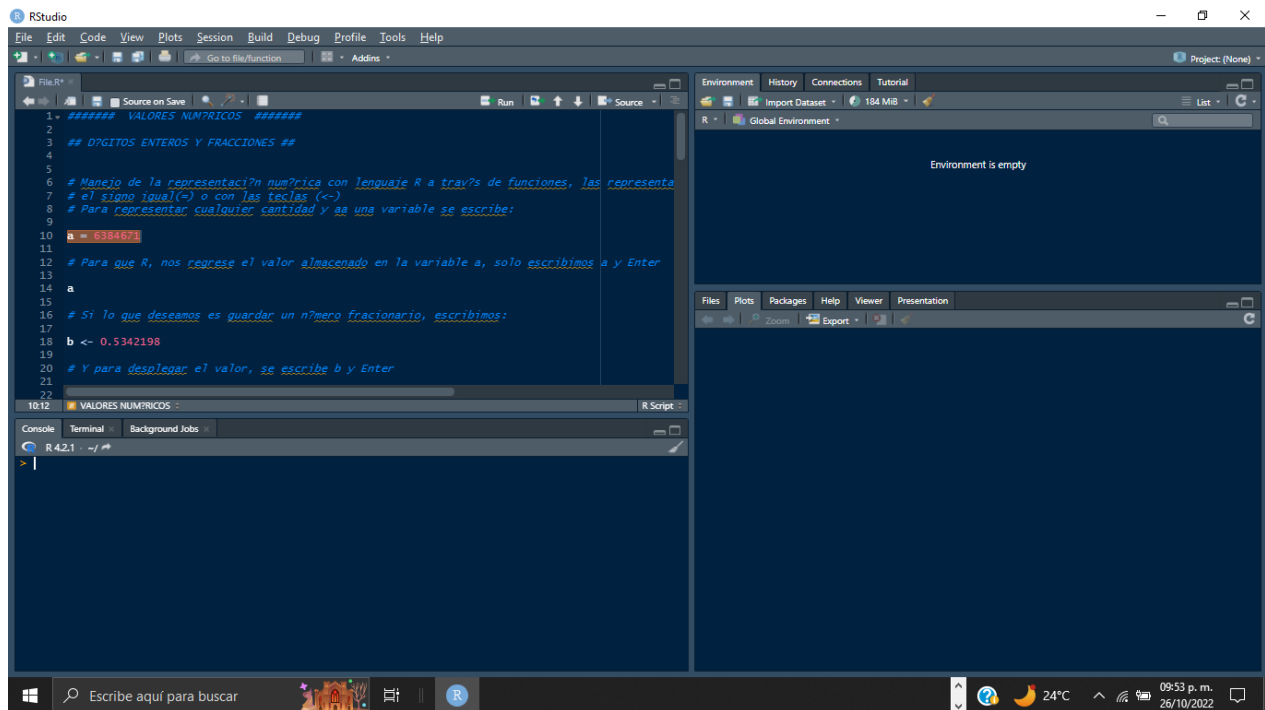


Abrimos el archivo descargado.

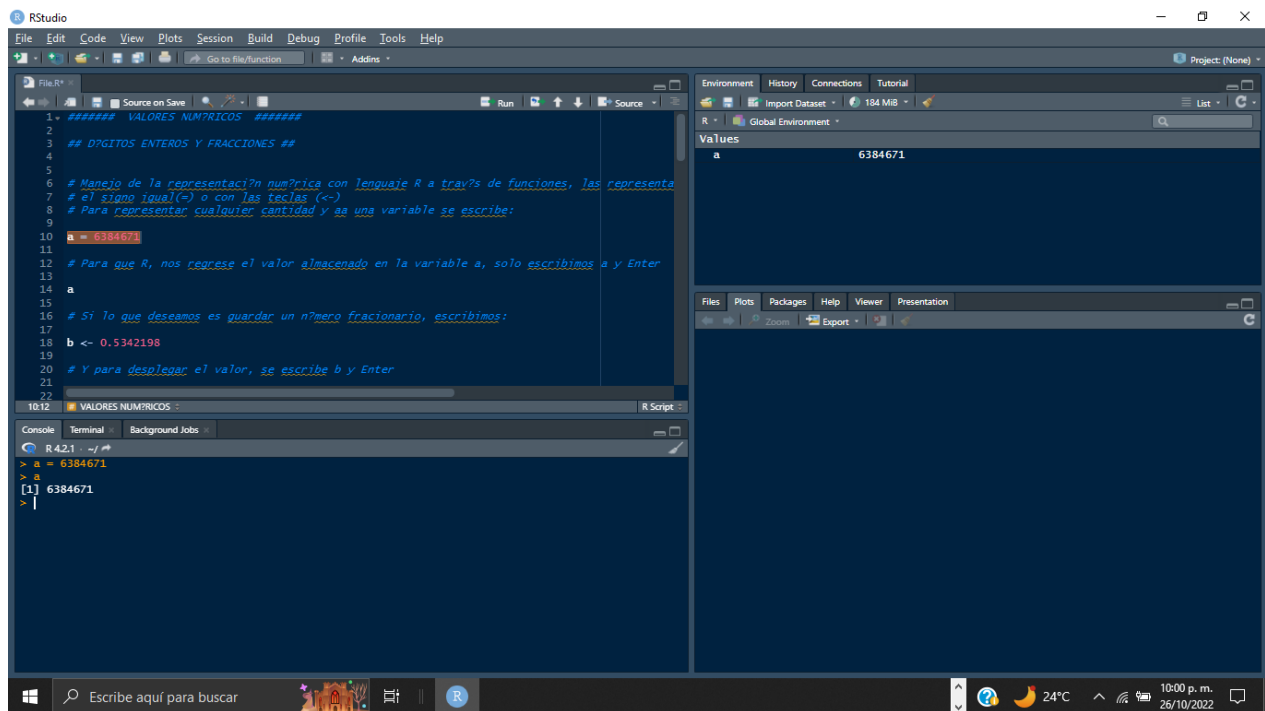


Y cargamos los valores_numericos.R

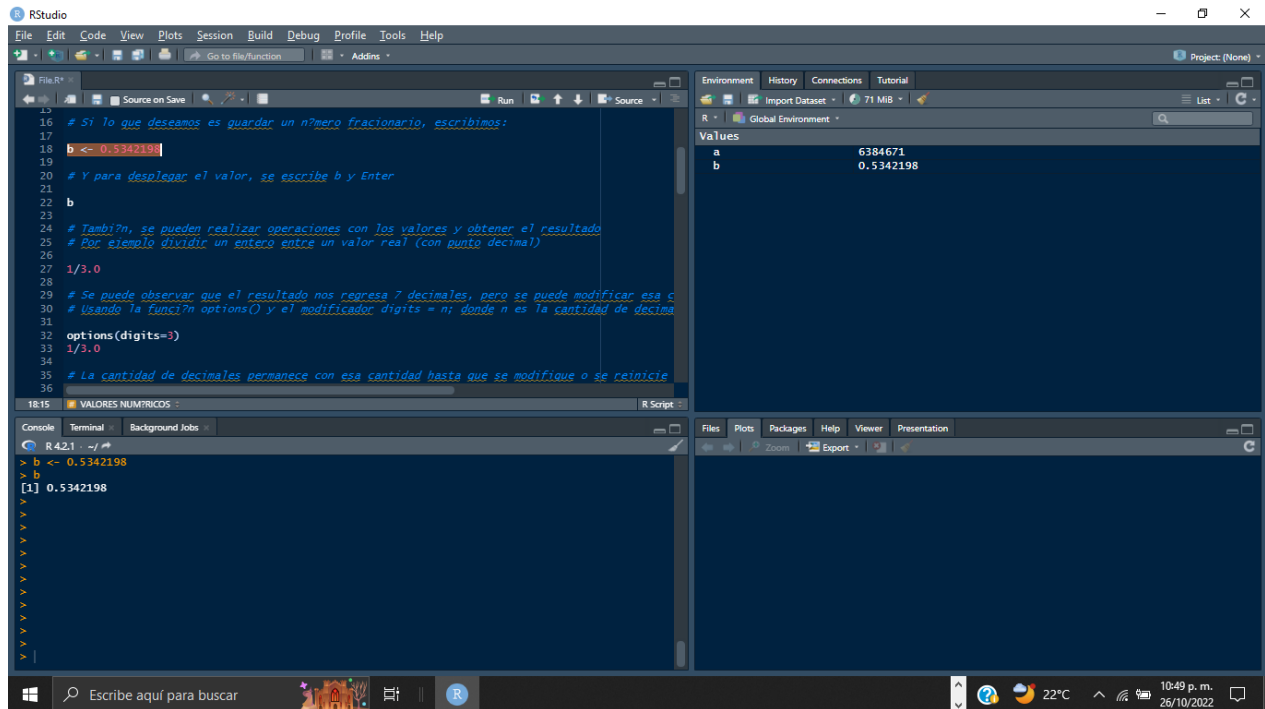
Ejecución de valores_numéricos.R



Ejecutamos el primer script en donde se asigna el valor de 6384671 a la letra “a” con el signo =.



Se comprueba que al escribir la letra “a” nos regresa el valor almacenado que es 6384671.



Se ejecuta el segundo script en donde se almacena un numero fraccionario a cualquier letra, en este caso se utiliza la letra “b”, para que funcione se tiene que anteponer a la cifra los signos <- y se comprueba que al escribir la letra “b” nos regresa el valor asignado, en este caso es 0.5342198.

```
19 # Y para desplegar el valor, se escribe b y Enter
20
21
22 b
23
24 # Tambi n, se pueden realizar operaciones con los valores y obtener el resultado
25 # Por ejemplo dividir un entero entre un valor real (con punto decimal)
26
27 1/3.0
28
29 # Se puede observar que el resultado nos regresa 7 decimales, pero se puede modificar esa c
30 # Usando la funci n options() y el modificador digits = n; donde n es la cantidad de decima
31
32 options(digits=3)
33 1/3.0
34
35 # La cantidad de decimales permanece con esa cantidad hasta que se modifique o se reinicie
36
37 options(digits=7)
38 1/3.0
39
40
```

Environment: Global Environment

Variable	Value
a	6384671
b	0.5342198

Console: R 4.2.1

```
> 1/3.0
[1] 0.3333333
```

Al ejecutar el siguiente script se le indica al programa que divida un entero, con el signo /, a un valor real con punto decimal, mostrando como resultado el siguiente valor: 0.3333333, el cual se mantendr  hasta que delimitemos un nuevo valor de decimales, para reducir la cantidad de decimales ejecutaremos el siguiente script: options(digits=3), en donde le indicamos al programa solo mostrar 3 decimales.

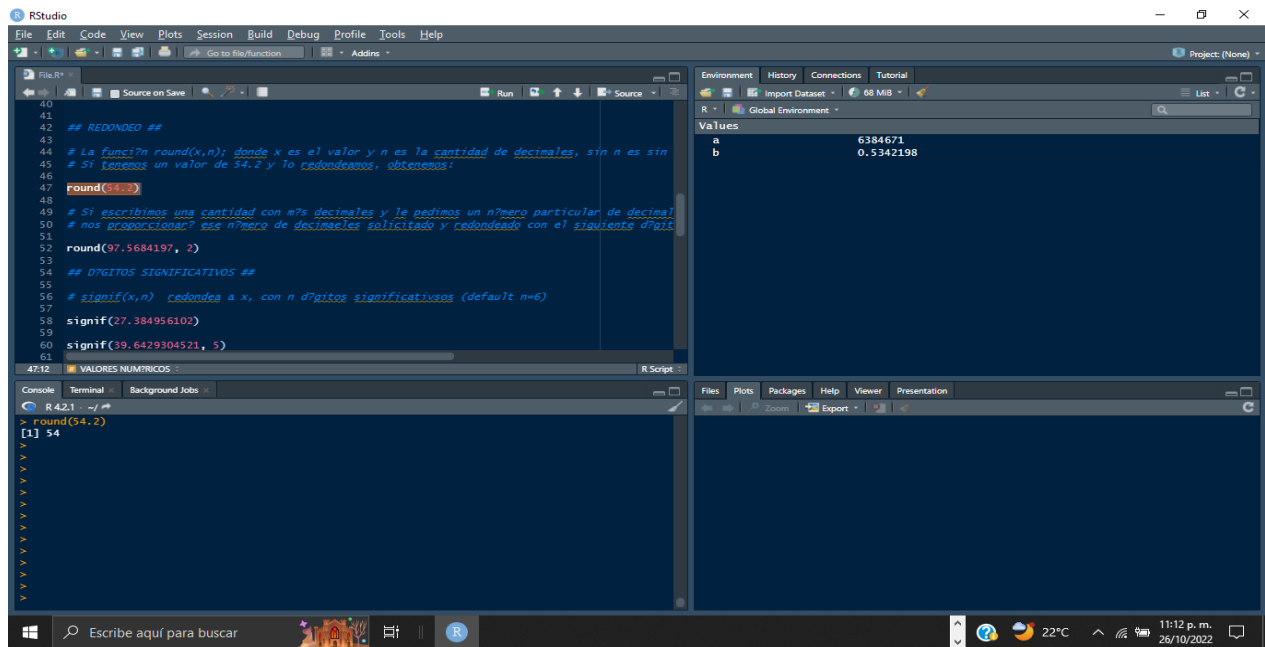
```
19 # Y para desplegar el valor, se escribe b y Enter
20
21
22 b
23
24 # Tambi n, se pueden realizar operaciones con los valores y obtener el resultado
25 # Por ejemplo dividir un entero entre un valor real (con punto decimal)
26
27 1/3.0
28
29 # Se puede observar que el resultado nos regresa 7 decimales, pero se puede modificar esa c
30 # Usando la funci n options() y el modificador digits = n; donde n es la cantidad de decima
31
32 options(digits=3)
33 1/3.0
34
35 # La cantidad de decimales permanece con esa cantidad hasta que se modifique o se reinicie
36
37 options(digits=7)
38 1/3.0
39
40
```

Environment: Global Environment

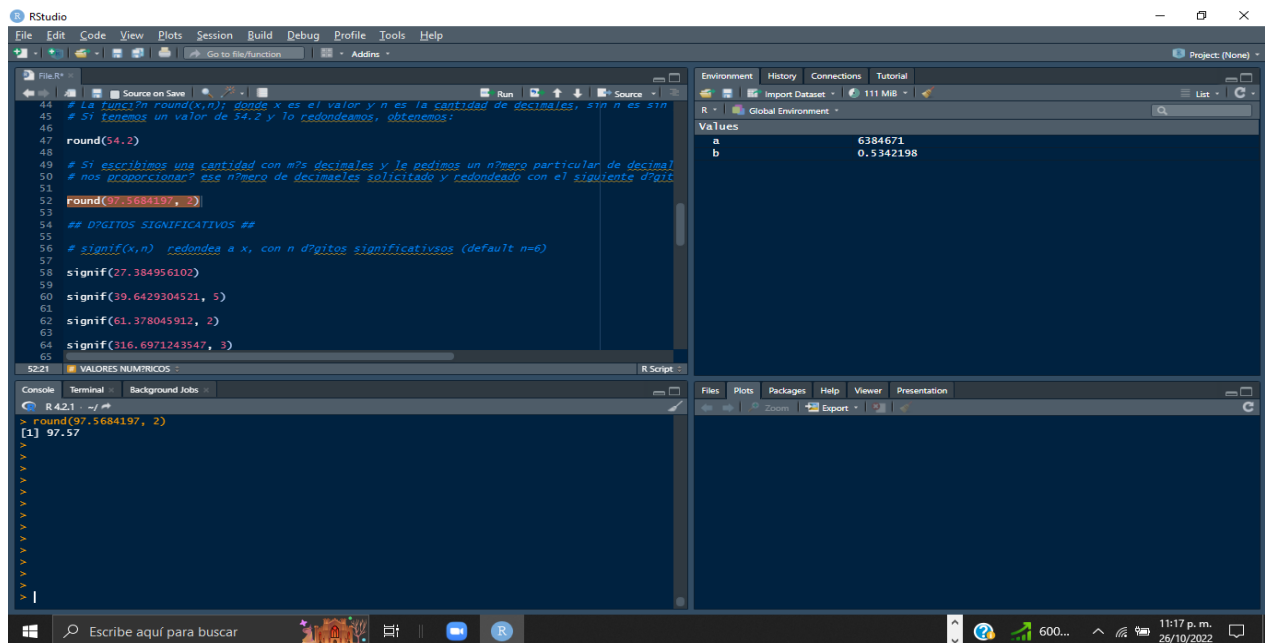
Variable	Value
a	6384671
b	0.5342198

Console: R 4.2.1

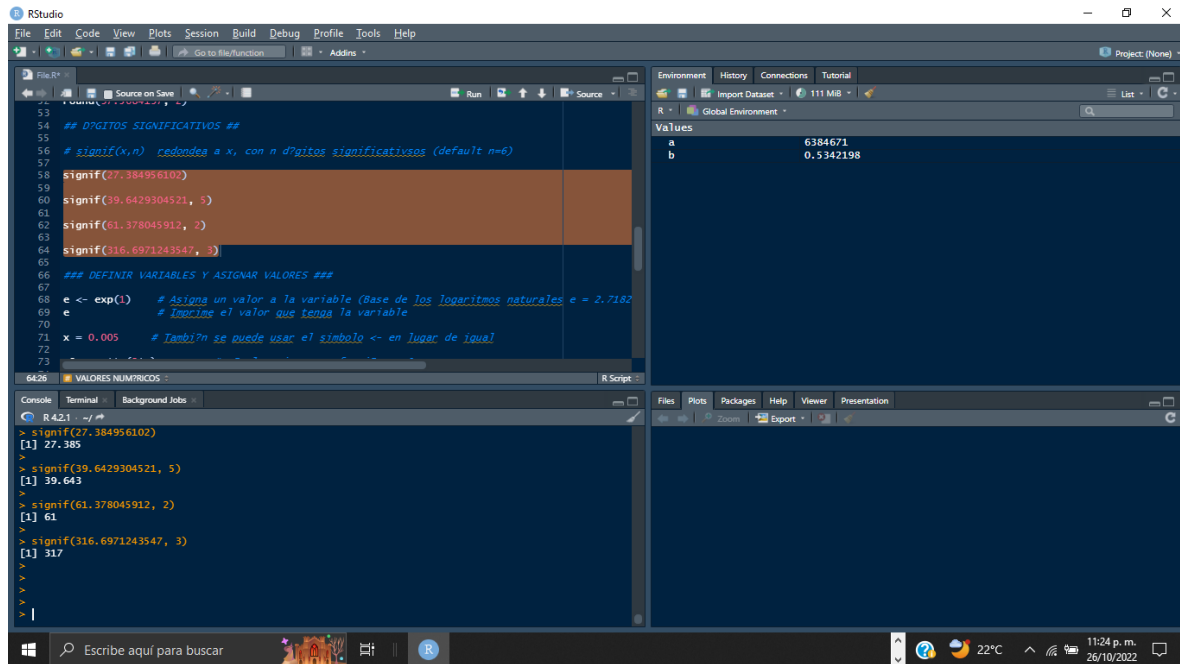
```
> options(digits=3)
> 1/3.0
[1] 0.333
```



En el siguiente script solicitaremos que redondee la cifra 54.2 con la función `round(x,n)`, una vez ejecutado el script nos devuelve el valor de 54, por estar debajo de la media. Si fuera 54.6 nos devolvería el valor de 55.



Al ingresar una cifra con un mayor número de decimales se puede condicionar en la función, indicando la cantidad de decimales a mostrar después de la cifra. Ejemplo: `round(97.5684197,2)`



En el siguiente script indicamos al programa redondear x número con n dígitos significativos utilizando la siguiente formula, `signif(x,n)`, en estos cuatro resultados en la primera cifra se muestra que el programa redondea solo los decimales, al tener como máximo 6 décimas, en el siguiente se condiciona con 5 números significativos, en la siguiente con 2 número significativos y en la última con 3 números significativos, redondeando la cifra al siguiente número, si las decimas están por encima de la media.

The screenshot displays the RStudio environment. The main editor window contains an R script with the following code:

```
## DEFINIR VARIABLES Y ASIGNAR VALORES ##
e <- exp(1) # Asigna un valor a la variable (Base de los logaritmos naturales e = 2.718)
# Imprime el valor que tenga la variable
e
x <- 0.005 # También se puede usar el símbolo <- en lugar de igual
x0 = e ** (2*x) # Se le asigna una función a x0
tex = "El valor de x0 es: " # A la variable tex, se le asigna una cadena
cat(tex, x0) # Se obtiene los resultados con la instrucción cat, que concatena y
# Otro ejemplo:
x0 = 1
x1 = x0 - pi + x0 + 1
x1
```

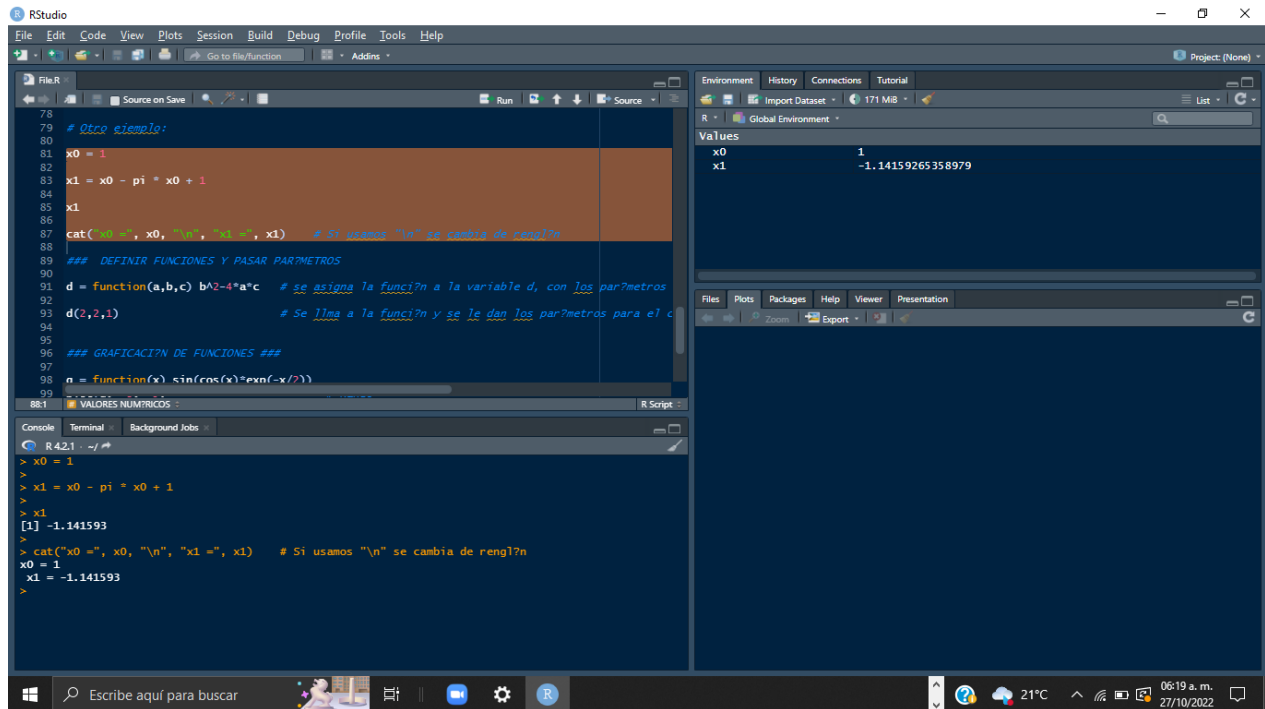
The console window at the bottom shows the execution of the script, with the following output:

```
> e <- exp(1) # Asigna un valor a la variable (Base de los logaritmos naturales e = 2.718281)
> e
[1] 2.718282
> x <- 0.005 # También se puede usar el símbolo <- en lugar de igual
> x0 = e ** (2*x) # Se le asigna una función a x0
> tex = "El valor de x0 es: " # A la variable tex, se le asigna una cadena
> cat(tex, x0) # Se obtiene los resultados con la instrucción cat, que concatena y
onvierte a string
El valor de x0 es: 1.01005
>
```

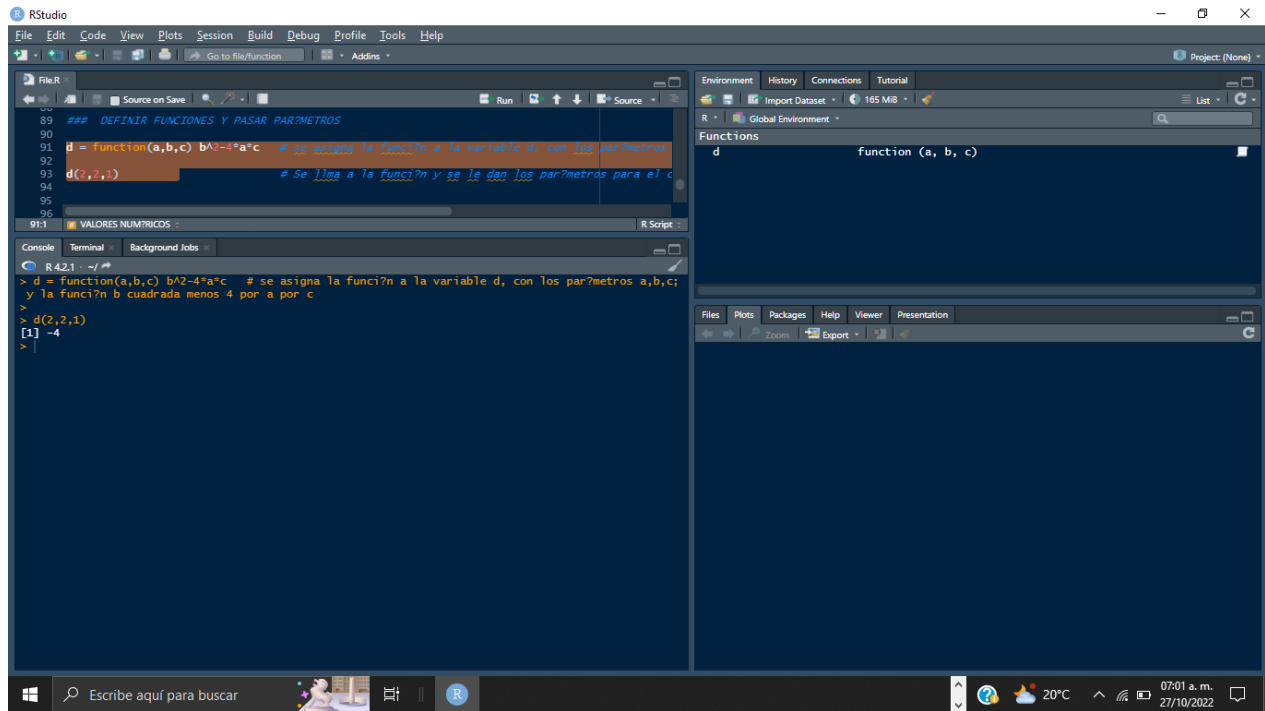
The Environment pane on the right shows the current variables in the Global Environment:

Variable	Value
e	2.71828182845905
tex	"El valor de x0 es: "
x	0.005
x0	1.01005016708417

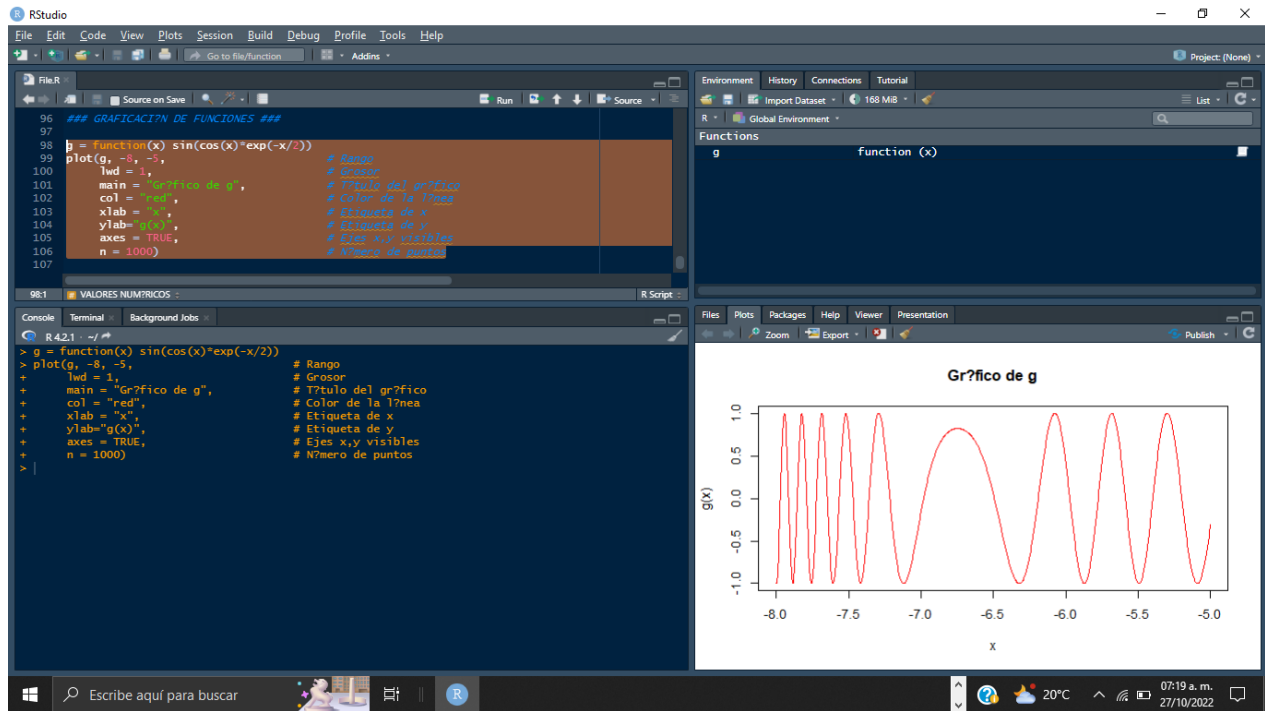
En el siguiente script se asigna un valor a la letra “e”, donde “e” es igual a, 2.718281 tomando la base de los logaritmos naturales, quedando de la siguiente forma `e <- exp (1)`, para la letra “x” se le asigna el valor de 0.005 utilizando los mismos signos asignando una función, donde $x0=e^{2x}$ lo que es igual a $x0=e^{2 \times 0.005}$, con el comando text se le asigna una cadena de texto a la variable entre comillas, que nos mostrara en pantalla el texto “ El valor de x0 es: “, y por ultimo pedimos al programa realice la operación con el comando `cat (text, x0)`, que nos mostrara el resultado final de la operación, imprimiendo en pantalla el texto y el resultado de la operación.



En el siguiente ejemplo al ejecutar el script asignamos nuevamente un valor a la variable, x0, asignamos la función para encontrar el valor de x1 quedando de la siguiente forma: $x1 = x0 - \pi * x0 + 1$ lo que es igual, $x1 = x0 - \pi(x0 + 1)$ y por ultimo pedimos al programa que realice la operación con el comando cat solicitando separe los resultados de x0 y de x1 con un salto de renglón, indicándolo con “\n” quedando de la siguiente forma: `cat("x0 =", x0, "\n", "x1 =", x1)`.

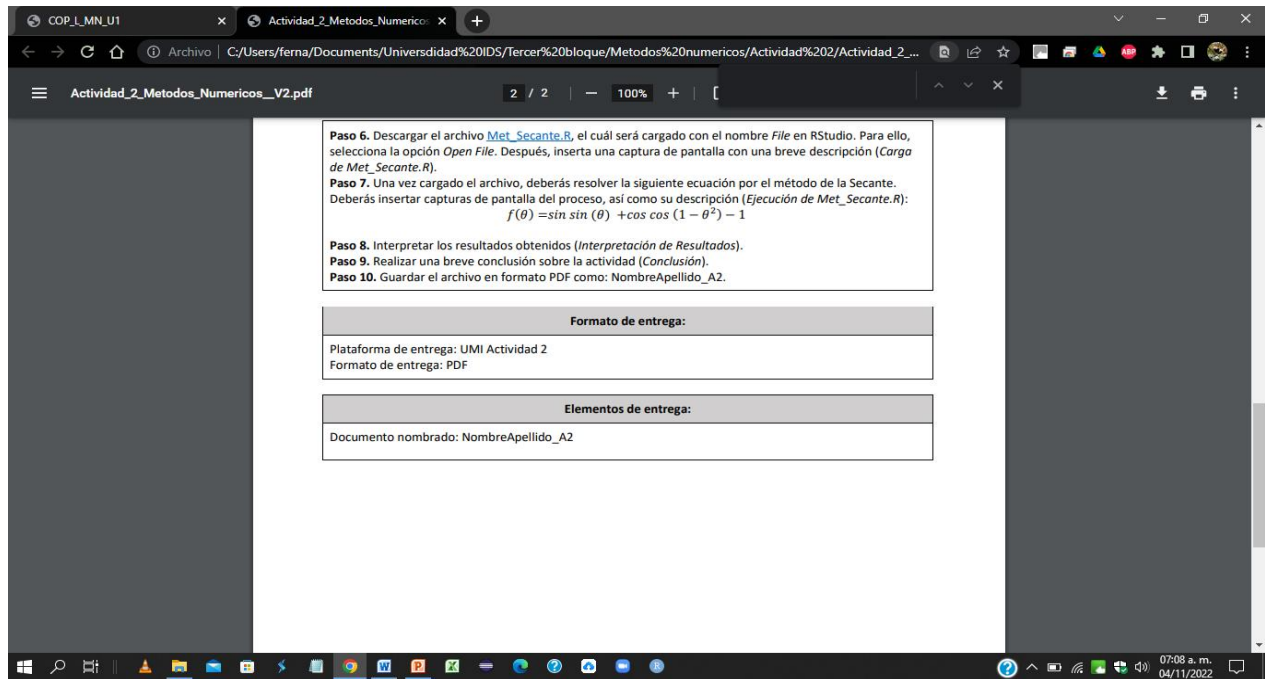


En el siguiente script se definirá una función y se pasaran parámetros a una variable, en este caso se asigna la función a la letra “d” con los parámetros (a,b,c), donde “a=2, b=2 y c=1”, y la función b cuadrada, menos 4, multiplicada por “a” y por “c” quedando de la siguiente forma: $d = \text{function}(a,b,c) \ b^{2-4ac}$, indicándolo en el programa como $d = \text{function}(a,b,c) \ b^{2-4*a*c}$, dando como resultado -4.

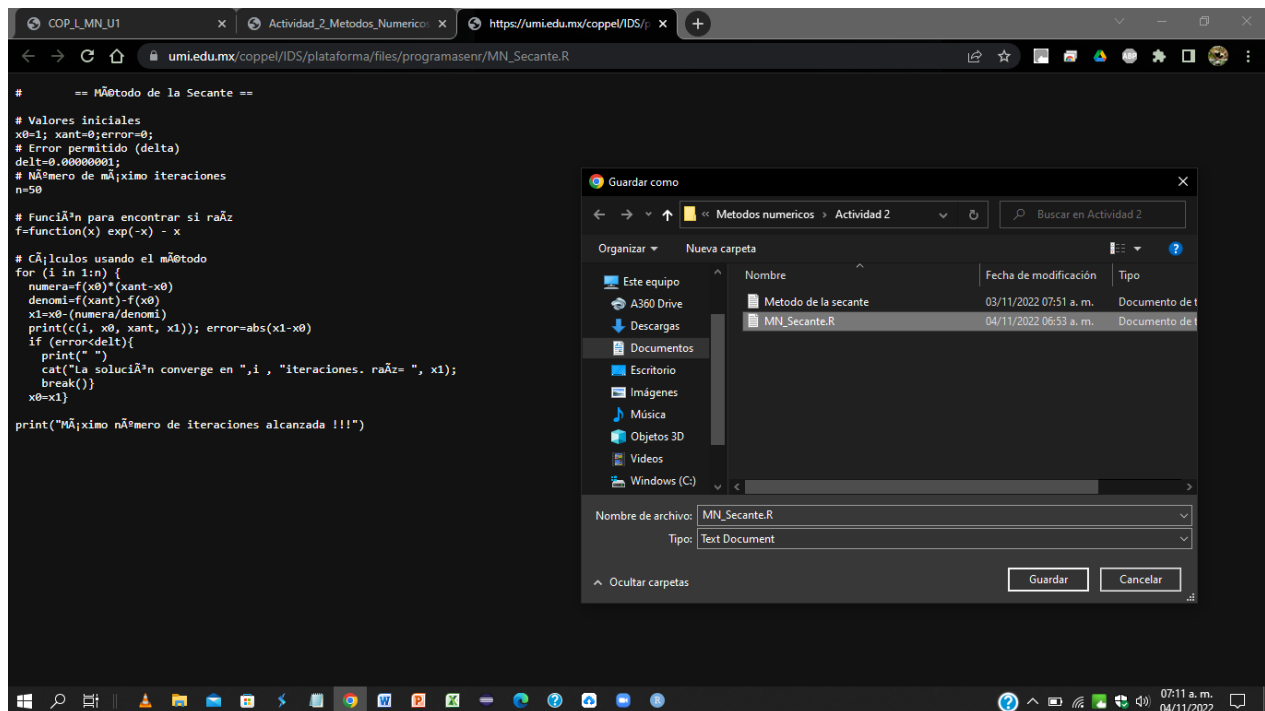


Por ultimo al ejecutar el siguiente script nos permitirá graficar cualquier función, indicando el rango, grosor, título del gráfico, color de la línea, etiqueta de “x” y etiqueta de “y”, mostrando los ejes visibles de “x” y “y”, y el número de puntos.

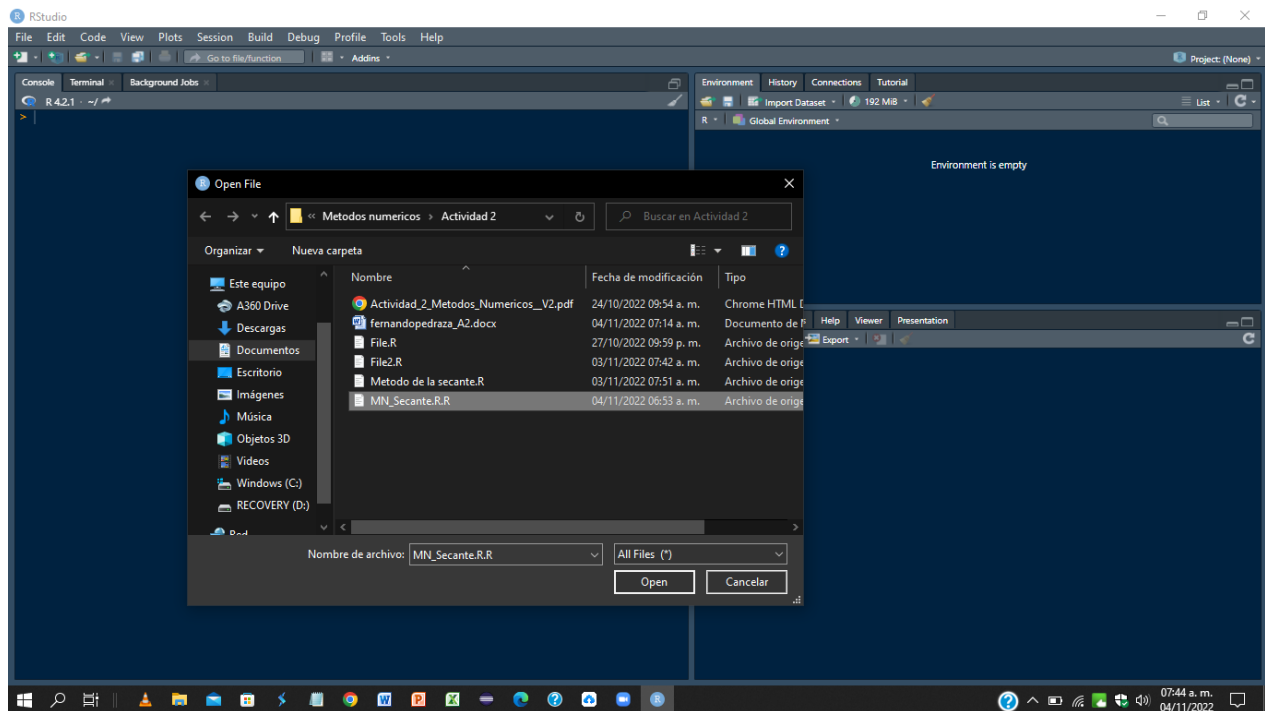
Carga de Met_Secante.R



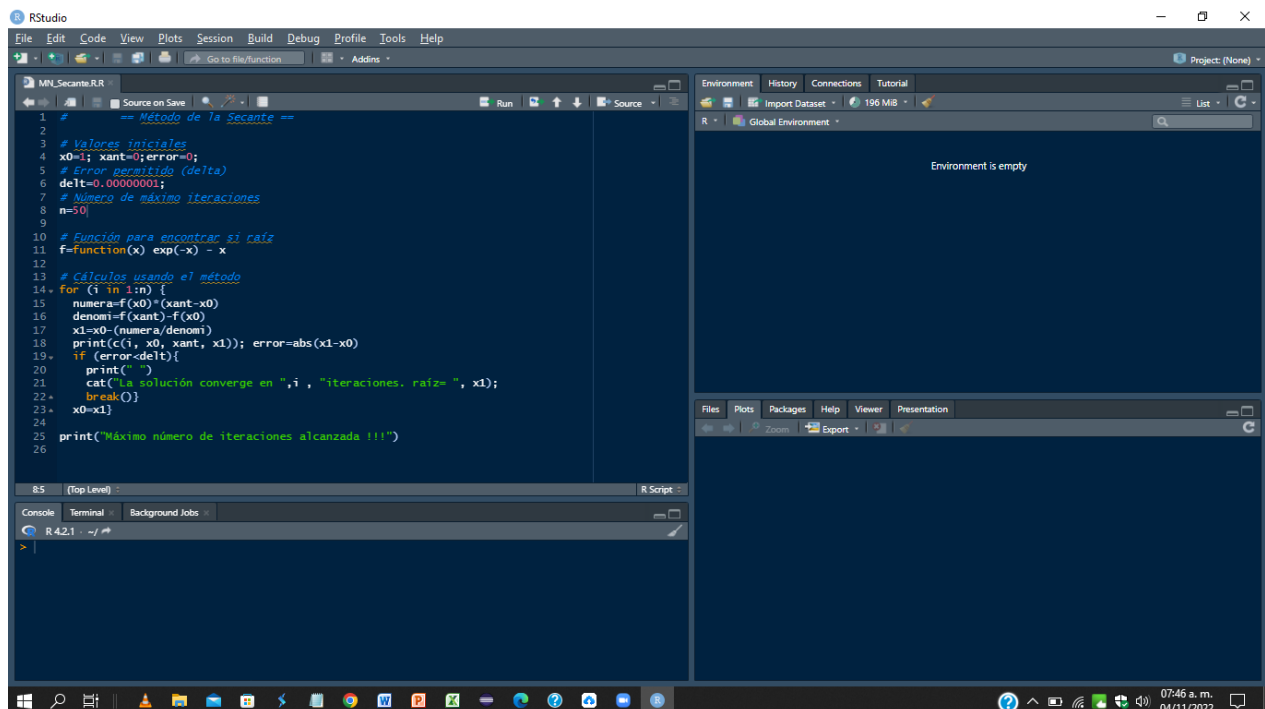
Abrimos el enlace del archivo de Met_Secante.R



Guardamos el archivo con el nombre MN_Secante-R.

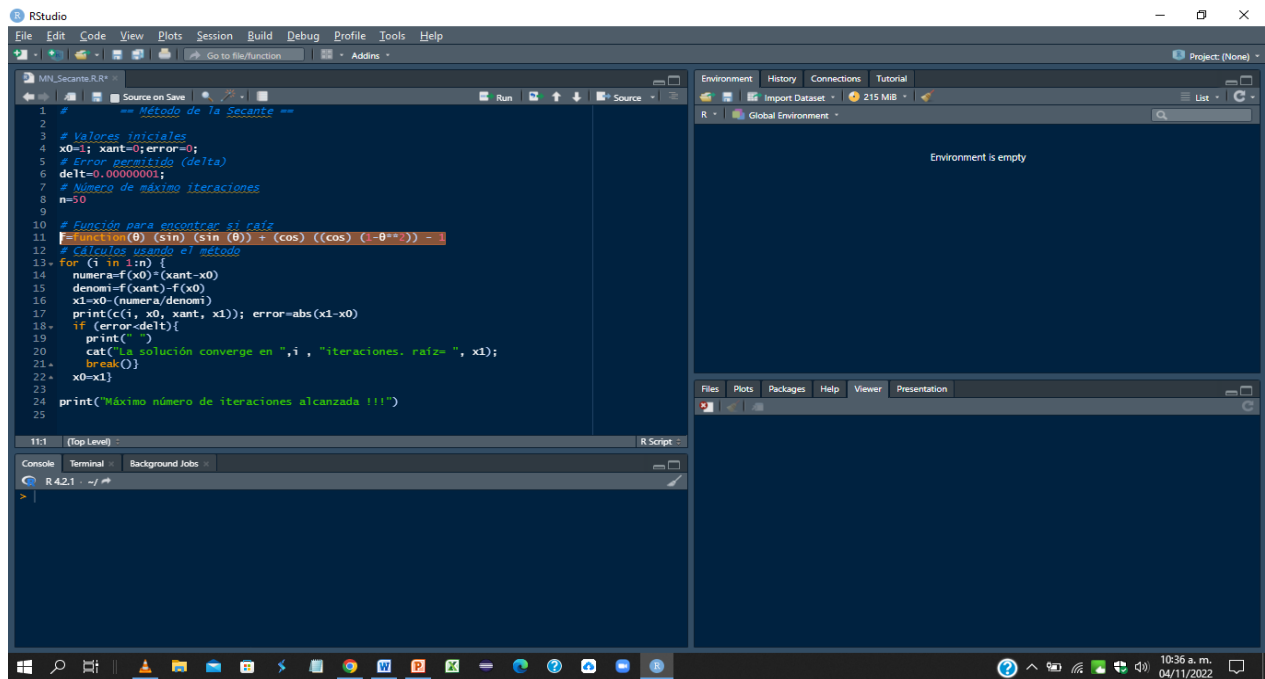


Abrimos R Studio y cargamos el archivo MN_Secante.R



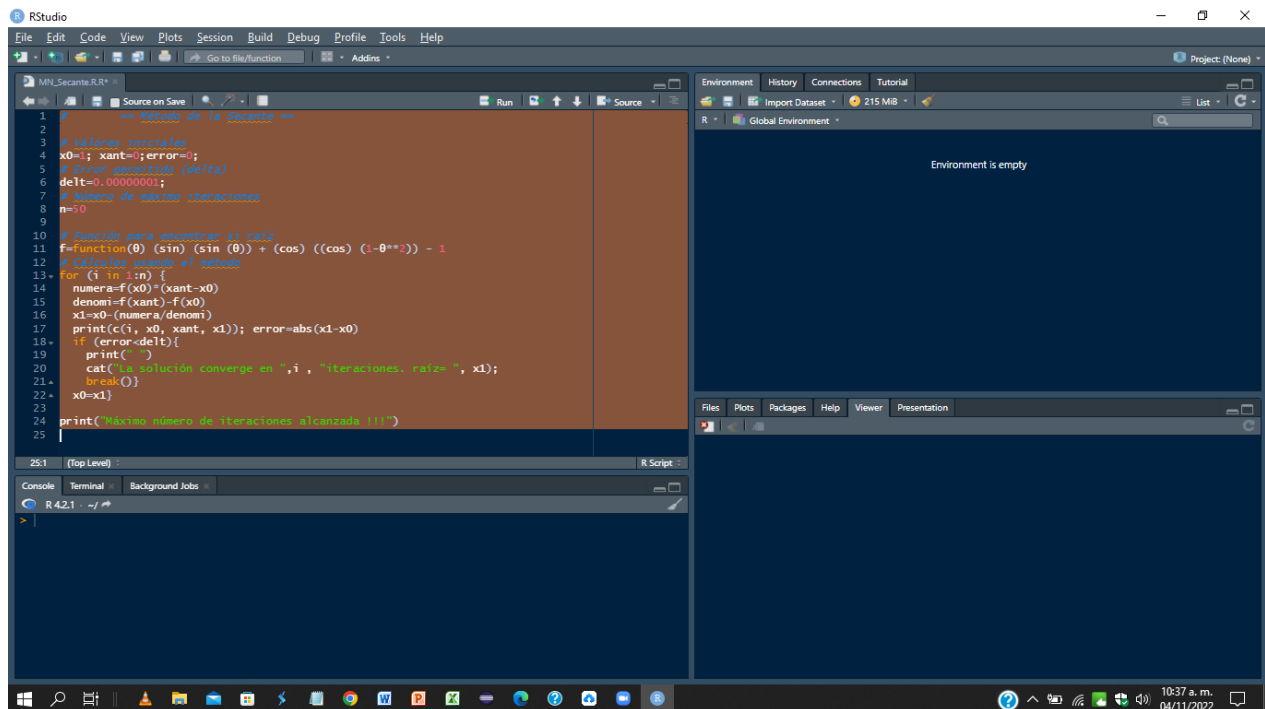
Archivo Met_Secante.R cargado en R Studio

Ejecución de Met_Secante.R



```
1 # == Método de la Secante ==
2
3 # Valores iniciales
4 x0=1; xant=0; error=0;
5 # Error permitido (delta)
6 delt=0.00000001;
7 # Número de máximas iteraciones
8 n=50
9
10 # Función para encontrar la raíz
11 f=function(theta) (sin(sin(theta)) + (cos(cos(1-theta**2))) - 1
12 # Cálculos usando el método
13 for (i in 1:n) {
14   numera=f(x0)-(xant-x0)
15   denomi=f(xant)-f(x0)
16   x1=x0-(numera/denomi)
17   print(c(i, x0, xant, x1)); error=abs(x1-x0)
18   if (error<delt){
19     print(" ")
20     cat("La solución converge en ", i , " iteraciones. raíz= ", x1);
21     break()
22   }
23   x0=x1
24 }
25 print("Máximo número de iteraciones alcanzada !!!")
```

Cambiamos la función por la función solicitada de $f(\theta) = \sin \sin (\theta) + \cos \cos (1- \theta^2) - 1$



```
1 # == Método de la Secante ==
2
3 # Valores iniciales
4 x0=1; xant=0; error=0;
5 # Error permitido (delta)
6 delt=0.00000001;
7 # Número de máximas iteraciones
8 n=50
9
10 # Función para encontrar la raíz
11 f=function(theta) (sin(sin(theta)) + (cos(cos(1-theta**2))) - 1
12 # Cálculos usando el método
13 for (i in 1:n) {
14   numera=f(x0)-(xant-x0)
15   denomi=f(xant)-f(x0)
16   x1=x0-(numera/denomi)
17   print(c(i, x0, xant, x1)); error=abs(x1-x0)
18   if (error<delt){
19     print(" ")
20     cat("La solución converge en ", i , " iteraciones. raíz= ", x1);
21     break()
22   }
23   x0=x1
24 }
25 print("Máximo número de iteraciones alcanzada !!!")
```

Seleccionamos para ejecutar el programa en base al método de la secante y su función:

$$x_y = x_a - \frac{f(x_a) \cdot (x_b - x_a)}{f(x_b) - f(x_a)}$$

```

1 # Función para encontrar la raíz
2 x0=1; xant=0; error=0;
3 # Error permitido (delta)
4 delt=0.00000001;
5 # Número de Máximo Iteraciones
6 n=50
7
8 # Función para encontrar la raíz
9 f=function(x) (sin(x) + (cos((1-0**2))) - 1)
10 # Método de la secante
11 for (i in 1:n) {
12   numera=f(x0)*(xant-x0)
13   denomi=f(xant)-f(x0)
14   x1=x0-(numera/denomi)
15   print(c(i, x0, xant, x1)); error=abs(x1-x0)
16   if (error<delt){
17     print(" ")
18     cat("La solución converge en ", i, " iteraciones. raíz=", x1);
19     break()
20   }
21   x0=x1
22 }
23
24 print("Máximo número de iteraciones alcanzada !!!")

```

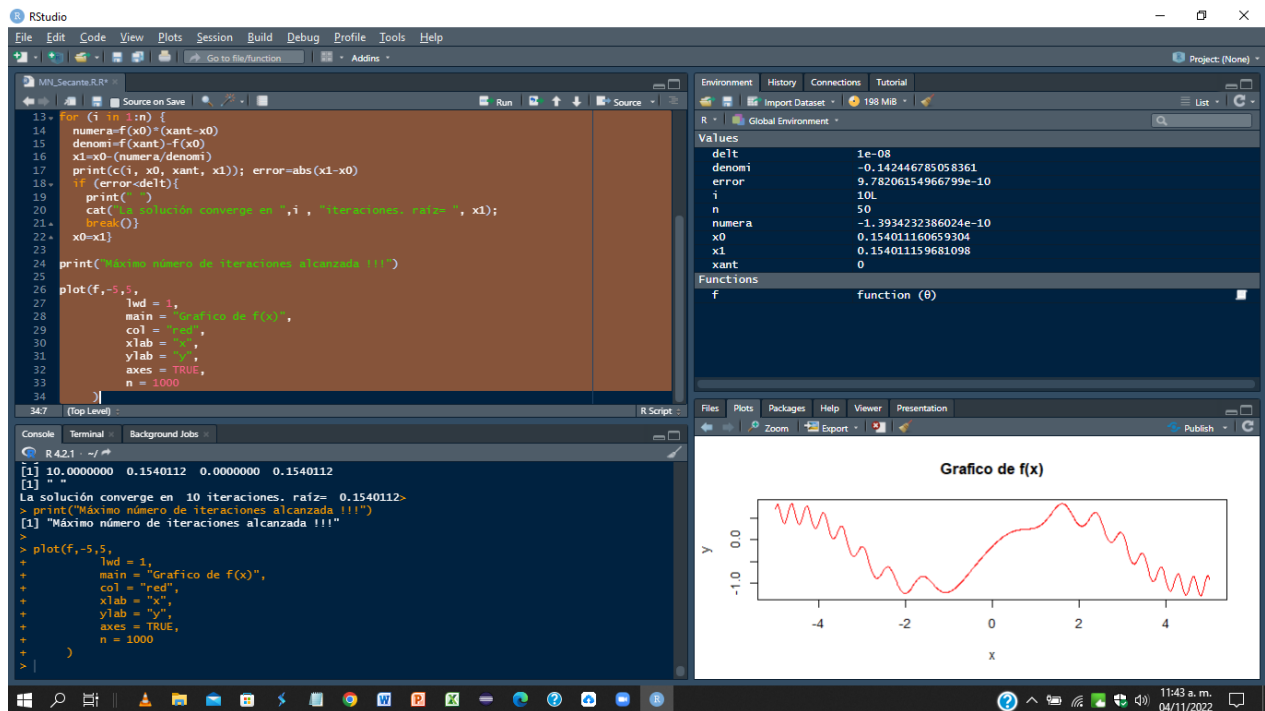
Console Output:

```

[1] 1.0000000 1.0000000 0.0000000 0.3325296
[1] 2.0000000 0.3325296 0.0000000 0.1746435
[1] 3.0000000 0.1746435 0.0000000 0.1559341
[1] 4.0000000 0.1559341 0.0000000 0.1541861
[1] 5.0000000 0.1541861 0.0000000 0.1540270
[1] 6.0000000 0.1540270 0.0000000 0.1540126
[1] 7.0000000 0.1540126 0.0000000 0.1540113
[1] 8.0000000 0.1540113 0.0000000 0.1540112
[1] 9.0000000 0.1540112 0.0000000 0.1540112
[1] 10.0000000 0.1540112 0.0000000 0.1540112
[1] " "
La solución converge en 10 iteraciones. raíz= 0.1540112>
> print("Máximo número de iteraciones alcanzada !!!")
[1] "Máximo número de iteraciones alcanzada !!!"

```

Y se obtiene el resultado que indica que la raíz es de 0.1540112 y converge en 10 iteraciones



Obteniendo la siguiente gráfica.

Interpretación de resultados

```

3 # Valores iniciales
4 x0=1; xant=0;error=0;
5 # Paso de secante (delta)
6 delt=0.00000001;
7 # Numero de máximas iteraciones
8 n=50
9
10 # función para encontrar el raíz
11 f=function(θ) (sin(θ)sin(θ)) + (cos(θ)cos(1-θ²)) - 1
12 # Cálculos usando el método
13 for (i in 1:n) {
14   numera=f(x0)-(xant-x0)
15   denomi=f(xant)-f(x0)
16   x1=x0-(numera/denomi)
17   print(c(i, x0, xant, x1)); error=abs(x1-x0)
18   if (error<delt){
19     print(" ")
20     cat("La solución converge en ",i, " iteraciones. raíz=", x1);
21     break()
22   }
23   x0=x1
24 }
25 print(" Máximo número de iteraciones alcanzada !!!")

```

Console Output:

```

[1] 1.0000000 1.0000000 0.0000000 0.3325296
[1] 2.0000000 0.3325296 0.0000000 0.1746435
[1] 3.0000000 0.1746435 0.0000000 0.1559341
[1] 4.0000000 0.1559341 0.0000000 0.1541861
[1] 5.0000000 0.1541861 0.0000000 0.1540270
[1] 6.0000000 0.1540270 0.0000000 0.1540126
[1] 7.0000000 0.1540126 0.0000000 0.1540113
[1] 8.0000000 0.1540113 0.0000000 0.1540112
[1] 9.0000000 0.1540112 0.0000000 0.1540112
[1] 10.0000000 0.1540112 0.0000000 0.1540112
[1] " "
[1] "La solución converge en 10 iteraciones. raíz= 0.1540112"
> print("Máximo número de iteraciones alcanzada !!!")
[1] "Máximo número de iteraciones alcanzada !!!"

```

Plantilla clase 1.xlsx - Microsoft Excel

Iteraciones	xa	xb	f(xa)	f(xb)	x	f(x)	error
1	0	1	-0.14244678	0.28592645	0.33252961	0.12877873	
2	0	0.33252961	-0.14244678	0.12877873	0.1746435	0.01709113	0.904048001
3	0	0.1746435	-0.14244678	0.01709113	0.15593413	0.00161497	0.11998254
4	0	0.15593413	-0.14244678	0.00161497	0.15418607	0.00014708	0.011337334
5	0	0.15418607	-0.14244678	0.00014708	0.15402703	1.3349E-05	0.001032526
6	0	0.15402703	-0.14244678	1.3349E-05	0.1540126	1.2113E-06	9.37156E-05
7	0	0.1540126	-0.14244678	1.2113E-06	0.15401129	1.099E-07	8.50331E-06
8	0	0.15401129	-0.14244678	1.099E-07	0.15401117	9.9717E-09	7.71529E-07
9	0	0.15401117	-0.14244678	9.9717E-09	0.15401116	9.0475E-10	7.00028E-08
10	0	0.15401116	-0.14244678	9.0475E-10	0.15401116	8.2091E-11	6.35153E-09
11	0	0.15401116	-0.14244678	8.2091E-11	0.15401116	7.4483E-12	5.76291E-10
12	0	0.15401116	-0.14244678	7.4483E-12	0.15401116	6.7568E-13	5.22879E-11
13	0	0.15401116	-0.14244678	6.7568E-13	0.15401116	6.1284E-14	4.74334E-12
14	0	0.15401116	-0.14244678	6.1284E-14	0.15401116	5.5511E-15	4.3018E-13
15	0	0.15401116	-0.14244678	5.5511E-15	0.15401116	0	3.89271E-14

SECANTE=
$$x_y = x_a - \frac{f(x_a) \cdot (x_b - x_a)}{f(x_b) - f(x_a)}$$

En comparación con la tabla de Excel se muestra la coincidencia en resultados donde se está tocando el cero en 0.15401116 y en R Studio en 0.1540112 en la décima iteración.

Conclusión.

En conclusión el análisis de conceptos nos ayuda a encontrar soluciones aproximadas a situaciones o problemas complicados de una forma explícita, en donde al aplicar las matemáticas de forma adecuada se pueden encontrar dichas soluciones, utilizando algoritmos atreves de sistemas aritméticos simples.

Referencias.

https://es.wikipedia.org/wiki/An%C3%A1lisis_formal_de_conceptos