

Laboratorio 1: Empezar con R y RStudio

Curso - Taller R como herramienta para el manejo de datos forestales

Dr. Marco A. González Tagle

07/ 12 / 2021

Índice

Instrucciones de la práctica	4
Consideraciones básicas para R y Rstudio	4
Primer contacto con la consola de R	5



Objetivos de la práctica

- Empezar a utilizar R como calculadora científica.
- Comprender el diseño del panel de RStudio.

Instrucciones de la práctica

Esta práctica consiste de una sesión inicial en R (cerca de 30 minutos), la cual involucra un primer contacto con R y Rstudio.

Consideraciones básicas para R y Rstudio

Vamos a utilizar principalmente RStudio, pero debes saber que R no es lo mismo que RStudio. Este último es solo un entorno de desarrollo integrado (IDE). Inicie RStudio y ubique los siguientes paneles (o paneles); su posición variará según la configuración de personalización:

- Consola (ubicación predeterminada a la izquierda).
- Environment/History (ubicación predeterminada: tabulado en la parte superior derecha).
- File/Plots/Packages/Help (ubicación predeterminada: tabulado en la parte inferior derecha).

Para iniciar con esta actividad debes de abrir un R Script siguiendo las instrucciones de la figura 1.

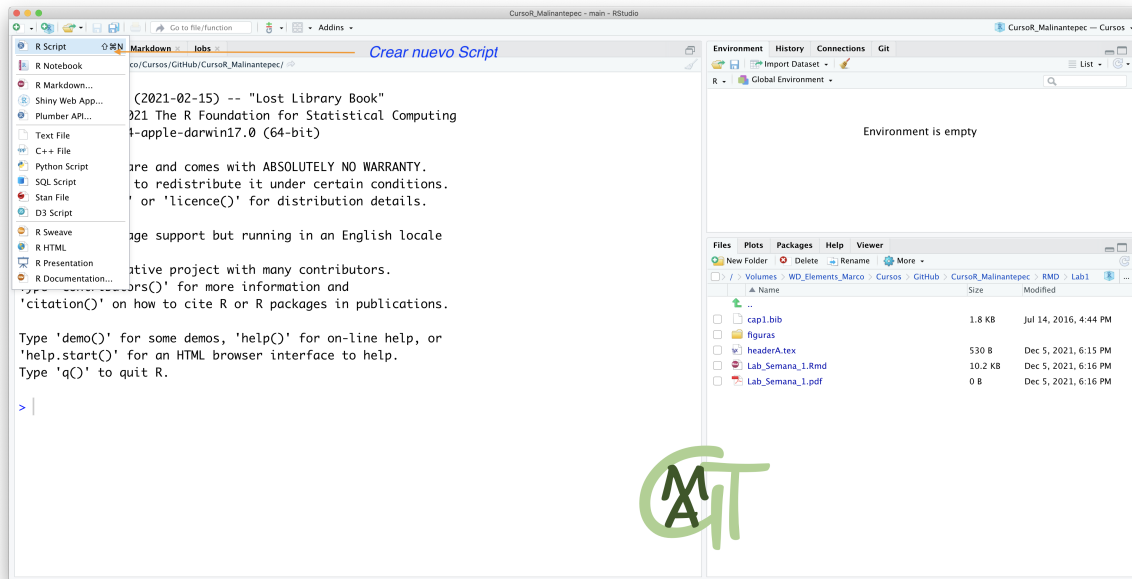


Figura 1: Iniciar un nuevo Script para ingresar nuestros datos.

Primer contacto con la consola de R

Comencemos a escribir cosas básicas en la consola, usando R como calculadora científica. Por ejemplo, considere las facturas mensuales de una estudiante universitaria de la MCF: teléfono celular \$ 300, transporte \$ 240, comestibles \$ 1527, gimnasio \$ 400, alquiler \$ 1500, otros \$ 1833. Puede usar R para encontrar los gastos totales de la estudiante:

```
# Gastos totales
300 + 240 + 1527 + 400 + 1500 + 1833
```

```
## [1] 5800
```

A menudo, será más conveniente crear objetos o variables que almacenen uno o más valores. Para hacer esto, escriba el nombre de la variable, seguido del operador de asignación `<-`, seguido del valor asignado. Por ejemplo, podrás crear un objeto llamado *celular* para los gastos de facturación de telefonía celular, después inspecciona el objeto creado.

```
celular <- 300
celular
```

```
## [1] 300
```

Todas las declaraciones de R en las que crea objetos, “asignaciones”, tienen esta forma:

```
objeto <- valor
```

esto significa que asigna un valor a un objeto dado; puede leer la asignación anterior como “celular se le asigna el valor de 300”.

Realiza más asignaciones para crear variables de transporte, comestibles, gimnasio, alquiler y otros con sus montos correspondientes.

- Ahora que tiene todas las variables, cree un objeto total con la suma de los gastos:
- Suponiendo que la estudiante tiene los mismos gastos todos los meses, ¿cuánto gastaría durante un semestre escolar? (suponga que el semestre implica cinco meses).
- Manteniendo la misma suposición sobre los gastos mensuales, ¿cuánto gastaría la estudiante durante un año escolar? (suponga que el año académico es de 10 meses).

Nombre de los objetos

Hay ciertas reglas que debe seguir al crear objetos y variables. Los nombres de los objetos no pueden comenzar con un dígito y no pueden contener otros caracteres como una coma o un espacio. Los siguientes ejemplos son correctos para el nombre de los objetos.

```
yo_uso_guion_bajo
otras.personas.usan.punto
OtrosUsanMayusculas
```

Los siguientes características en los nombres de los objetos no son aceptados (asignaciones invalidas):

```
# no puede iniciar con un número
5variable <- 5
# no puede iniciar con guión bajo
_invalid <- 10
# no puede contener una coma
my,variable <- 3
# no puede contener espacios
my variable <- 1
# Es correcto pero muy largo:
this_is_a_really_long_name <- 3.5
```

Funciones

R tiene muchas funciones [Crawley, 2007, Field et al., 2012, R Core Team, 2016]. Para usar una función debes escribir su nombre seguido de paréntesis. Dentro del paréntesis se establece una entrada. La mayoría de las funciones producirán algún tipo de salida:

```
# Valor absoluto (absolute value)
abs(10)
abs(-4)
# Raíz cuadrada (square root)
sqrt(9)
# Logaritmo natural (natural logarithm)
log(2)
```

Comentarios en R

Todos los lenguajes de programación usan un conjunto de caracteres para indicar que una parte o líneas de código específicas son comentarios, es decir, cosas que no se deben ejecutar. R usa el

símbolo de almohadilla o hashtag # para especificar comentarios. Cualquier código a la derecha de # no será ejecutado por R.

```
# este es un comentario
# este es otro comantario
2 * 9
4 + 5 # también se puede colocar un comentario
```

R distingue mayúsculas y minúsculas

R puede distinguir entre mayúsculas y minúsculas. Esto significa que *celular* no es lo mismo que *Celular* o *CELULAR*.

```
# Detectar matúsculas y minúsculas
celular <- 300
Celular <- -300
CELULAR <- 8000
```

```
celular + Celular
```

```
## [1] 0
```

ó el siguiente caso

```
CELULAR - celular
```

```
## [1] 7700
```

Obtener ayuda

Debido a que trabajamos con funciones todo el tiempo, es importante conocer ciertos detalles sobre cómo usarlas, qué entrada (s) se requiere y cuál es la salida devuelta. Hay varias formas de obtener ayuda. Si conoce el nombre de una función que le interesa conocer más, puede utilizar el *function help ()* y escribe el nombre de la función que estás buscando:

```
# Documentación acerca de la función abs
help(abs)

# Documentación acerca de la función mean
help(mean)
```

Alternativamente, puede usar un atajo usando el signo de interrogación? seguido del nombre de la función:

```
# Documentación acerca de la función abs
?abs

# Documentación acerca de la función mean
?mean
```

Cada función tiene su manual, por lo tanto, Comprender y aprender a leer la documentación del manual (es decir, ayuda) es de suma importancia.

– Title – Description – Usage of function – Arguments – Details – See Also – Examples!!!

la función `help()` solo funciona si conoces el nombre de la función que se está buscando. A veces, sin embargo, no se conoce el nombre, pero es posible que se conozca algunas palabras clave. Para buscar funciones relacionadas asociadas a una palabra clave, usa `help.search()` o simplemente doble `??`.

```
# buscar absolute
help.search("absolute")

# Alternativamente
??absolute
```

Autoevaluación

- Toma los objetos creados de la estudiantes (i.e variables) *transporte*, *comestibles*, *gimnasio*, *alquiler* y *otros* y escribamos dentro de la función de combinación `c()` para crear un vector llamado *gastos*:

```
gastos <- c(celular, transporte, comestibles, gimnasio, alquiler, otros)
```

- Ahora, use la función gráfica `barplot()` para producir un diagrama de barras de gastos:

```
barplot(gastos)
```

Referencias

Michael J. Crawley. *The R Book*. Wiley Publishing, 1st edition, 2007. ISBN 0470510242, 9780470510247.

Andy P. Field, Jeremy Miles, and Zoë Field. *Discovering Statistics Using R*. SAGE Publications, Thousand Oaks, CA, 2012. ISBN 9781446200469.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL <https://www.R-project.org/>.