



ICO 187 ANÁLISIS DE DATOS

CLASE 24: INTRODUCCIÓN A LA PROGRAMACIÓN EN R USANDO RSTUDIO

Año: 2021
Profesor: Sebastián Egaña

1. Introducción

Considere la necesidad de haber visto los videos subidos por el profesor en la semana AA.

Para poder realizar esto independiente del software, usaremos Rstudio Cloud, para esto necesitarán tener una cuenta en dicha página:

[Enlace acá](#)

- Trabajaremos vectores, matrices, lista y dataframes.

2. Aspectos previos

1. RStudio corresponde al GUI (Graphical User Interface) en base al lenguaje de programación R.
2. RStudio, consta de 4 partes: Script, Consola, Ambiente (Environment) y Explorador (Files, Plots, Packages, Help y Viewer).

Dentro del script se pueden abrir múltiples archivos relacionados con la edición de texto.

3. La interacción entre cada una de las partes de RStudio, se da de la siguiente manera:

R como lenguaje de programación más reciente, no se relaciona de manera directa con el lenguaje que “entienden” las computadores (lenguaje binario), por lo tanto, todo lo desarrollado en el Script debe ser traducido a través de la consola a lenguaje computacional para ser ejecutado por la computadora.

La interacción se puede dar a través de seleccionar la línea de código a ser leída, y presionar Ctrl + Enter. Otra forma es con el mouse dar click derecho y seleccionar RUN. También existe un boton en donde se puede seleccionar RUN.

4. R viene con una librería de comandos básica, que se denomina “base”. Existen otras librerías que pueden ser utilizadas, con comandos más avanzados para realizar operaciones y cálculos. En caso de ser la primera vez en que se quiera utilizar una nueva librería, esta debe ser instalada. Ya estando instalada, se le debe llamar utilizando el comando library, para habilitar los comandos provenientes de dicha librería. Los comandos deberían ser los siguientes:

```
install.packages("ggplot2")  
  
library("ggplot2")
```

5. En algunos casos, existe la necesidad de realizar comentarios en los Scripts, para esto se debe anteponer # a la línea para realizar el comentario. En otros casos podemos querer evitar que un código sea leído, utilizando # para esta finalidad.
6. Existen dos formas de operar en R, considerando la interacción entre Script y Consola: Generando los mismos outputs (salidas) en la consola o asignando dichas operatorias a una variable y luego generar una salida de la variable que contiene el valor almacenado de la variable.

La asignación de algún valor o múltiples valores a una variable en R se realiza mediante la utilización del signo = o signos <- y ->. Existen algunas diferencias sobre dichos operadores, veamos esto:

```
a = 1  
  
a
```

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

```
b <- 2  
  
b
```

```
## [1] 2
```

```
3 -> c  
  
c
```

```
## [1] 3
```

En el caso del signo =, lo de la derecha se asigna a la izquierda. Las “flechitas”, consideran la direccionalidad en la asignación.

3. Ayuda en R-Studio

Parte importante de programar, se relaciona con la manera en buscar cómo hacer algo, o cómo resolver algún problema. Para esto, no es necesario que inventemos la rueda, menos si estamos partiendo en la programación desde R.

1. Utilización del help (ayuda)

Se debe utilizar siempre anteponiendo el signo de pregunta; ya sea para búsquedas asociadas a comandos o paquetes. Esta ayuda estará disponible en la medida en que se tenga cargado el package correspondiente.

```
?sample
```

2. Cran y documentos relacionados

CRAN: Comprehensive R Archive Network

- Manual del package o a través de R Documentation:

[Link](#)

[Link](#)

3. Cheat Sheet

- Cheat Sheet: En algunos casos, existen estos documentos que muestran de manera visual la utilización de algunos comandos relacionados con alguna librería.

[Link](#)

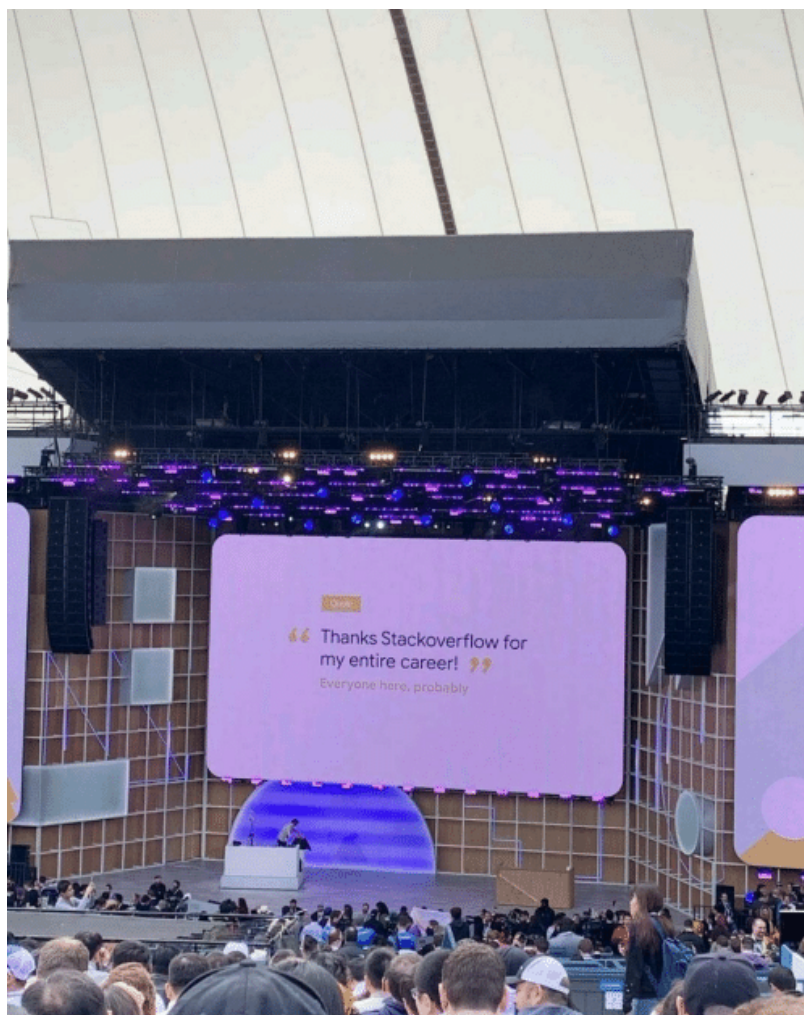
4. Páginas, Foros y Libros

Con una búsqueda simple del tipo

“How to do” “in R”.

Se encontrará material relacionado.

- Stackoverflow



Google I/O 2019

- Libros

R for Data Science

4. Programación y algoritmos

Un algoritmo se puede definir como un conjunto de instrucciones orientadas a solucionar un problema. En este sentido, es algo mucho más general que la programación. Pensemos en un problema trivial, y generemos el algoritmo relacionado.

Pensemos en la utilización de la guía telefónica para buscar el número de una persona. Sin necesidad de explicar mucho, la mejor manera de buscar un número es la siguiente: abrir el libro a la mitad, si el apellido que busco se encuentra en una u otra mitad, solo debo buscar en dicha mitad (se disminuye a la mitad el grupo de búsqueda). Luego la mitad que me queda, debo volver a dividirla en la mitad, y veo en que mitad esta la leta del apellido (se vuelve a dividir en la mitad), y así en adelante hasta encontrar el nombre.

Veamos esto como un algoritmo ¹.

1. Tomar la guía telefónica
2. Abrir la guía telefónica a la mitad
3. Buscar a la persona
4. Si la el apellido está en la página
5. —Llamar a la persona
6. Si la persona está en la primera mitad
7. —Abrir la primera mitad a la mitad
8. —Volver a la línea 3
9. Si la persona está en la segunda mitad
10. —Volver a la línea 3
11. En cualquier otro caso
12. —Salir

- Considerar que palabras como Tomar, Abrir, Buscar, Llamar que corresponde a acciones, serían lo que se denomina como funciones.
- La palabra “Si”, por lo general se asocia a condiciones que deben ser cumplidas para que algo pase. Lo que acompaña a dichos “si”, corresponden a expresiones booleanas las que en el caso de cumplirse ocurre algo y si no ocurre otra cosa.
- Líneas como Volver y Salir, son propias de construcciones llamadas “loops”, que permiten realizar cálculos de manera repetitiva o cíclica.

5. Hello World!

Un ejercicio básico es poder darle la instrucción al computador de imprimir (print) la frase “Hello World!” (Hola Mundo!). Veamos como hacer esto en R.

El siguiente código puede realizar la tarea en R:

```
print("Hello world!")
```

```
## [1] "Hello world!"
```

Veamos una estructura distinta, generando un objeto llamado print y después imprimiendolo en la consola:

¹Ejemplo en base al curso CS50 de Harvard

```
print <- print("Hello world!")
```

```
## [1] "Hello world!"
```

```
print
```

```
## [1] "Hello world!"
```

Ya vimos algo sobre esto, pero repasemoslo.

6. Pregunta 1²: Genere un vector que contenga la siguiente secuencia de números de 1 al 10.

Existen tres formas para poder realizar esto

- Forma 1. Esto es lo que ustedes escriben en el R Script

```
$seq(1,10,1) $
```

- Forma 2. Esto es lo que ustedes escriben en el R Script

```
$1:10 $
```

- Forma 3. Esto es lo que ustedes escriben en el R Script

```
seq(10)
```

- Forma 4. Esto es lo que ustedes escriben en el R Script

```
$c(1,2,3,4,5,6,7,8,9,10) $
```

Notas:

- Lo que escribas después de # se entiende como texto. ¡No es código!
- La sintaxis de la forma 1 contiene 3 argumentos:

```
seq(from = 1, to = 10, by = 1)
```

Veamos esto en detalle, aplicando posibles procesos de búsqueda de información para R.

¿Cómo podemos buscar información sobre la función seq?

7. Fechas Relevantes

Unidad	Evaluación	Ponderación	Fecha
Unidad I	Evaluación diagnóstica		25/03/2021
	Evaluación Individual Participación	(5 %)	05/04/2021
	Evaluación Grupal	(15 %)	27/04/2021 - 04/05/2021
	Evaluación Individual - Sumativa I	(30 %)	11/05/2021
Unidad II	Evaluación Grupal	(15 %)	27/05/2021
	Evaluación Grupal - Sumativa II	(15 %)	20/06/2021
Unidad III	Evaluación Formativa		27/06/2021
	Evaluación Grupal Sesión I- Sumativa III	(20 %)	08/07/2021
	Evaluación grupal Sesión II- Sumativa III	(20 %)	13/07/2021

²Preguntas en base al curso generado por Victor Macías para la Universidad de Chile