

# Laboratorio 1 - Introducción a R

## Métodos Cuantitativos

Carlos Eduardo Molina Berumen

El Colegio de México

3 de octubre de 2025



# Presentación del laboratorio

- Soy Carlos Molina, **CPRI22** en el CIDE.
- **¡IMPORTANTE!** email: carlos.e.molinaberumen@outlook.com

Objetivos de este taller:

- Aprenderemos a usar R.
- Aplicaremos y reforzaremos los conceptos que vemos en clase de Métodos Cuantitativos.

De manera rápida, díganme:

- Nombre y apellido (y cómo les gusta que se dirijan a ustedes)
- De dónde vienen
- RR. II. o PAP
- ¿Cómo se han sentido con la clase? ¿Hay temas que les gustaría reforzar?

- Nos veremos por seis sesiones
- Viernes de 13:30 a 15:00 hrs
- Salón:

R es un lenguaje de programación basado en objetos creado por estadísticos de Nueva Zelanda. Se llama R como una pequeña referencia a otro lenguaje de programación llamado “S”

Hay quienes encuentran más fácil programar y utilizar; no obstante, usaremos R studio porque es una forma más sencilla, más amigable y más interactiva de utilizar R.

# Primeros pasos

Para comenzar a programar, necesitamos dos cosas: Un proyecto y un script.

# ¿Script? ¿Proyecto? ¿Qué es eso?

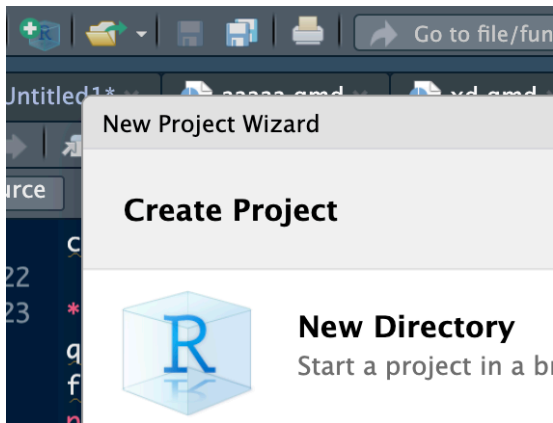
Todo lo que programamos en R lo guardamos usualmente en un script, y esos scripts van guardados en un proyecto.

El proyecto es la “carpeta madre” de todo lo que trabajes en R. En esa carpeta guardarás todo lo que precisas para alguna tarea (bases de datos, imágenes, codebooks, **SCRIPTS** etc.)

El *Script* es un archivo .R. Es el equivalente al .docx de Word, o al .ppt de PowerPoint. En este script guardarás tu código.

- Crear un proyecto: Esquina superior izquierda, click en la R dentro de una caja traslúcida. Escojan “New project”

```
knitr::include_graphics("Proyecto.png")
```





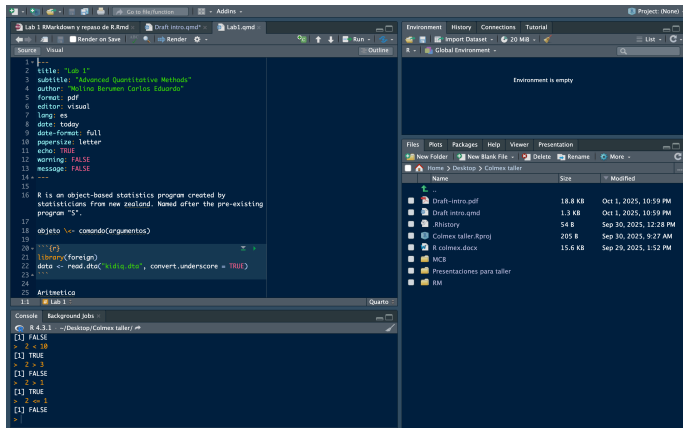
- New directory: sigan los mismos pasos para crear una carpeta normal. Escojan un path donde quieran que esté su carpeta. Nombren a su proyecto “Lab 1”.
- La esquina superior derecha indica que ya están dentro de su proyecto. Pueden acceder a él siguiendo el path que escogieron en cualquier otro momento.

Ya que estamos dentro de nuestro proyecto, es momento de crear nuestro primer script.

- Opción 1: Shift + Command + N en Mac, Ctrl + Shift + N en Windows
- Opción 2: Cuadrado blanco en la esquina superior izquierda, luego "R Script"

# ¿Qué es cada cosa? ¿Por qué tengo cuatro pantallas?

```
knitr::include_graphics("Pantalla R.png")
```



# ¿Qué es cada cosa? ¿Por qué tengo cuatro pantallas?

**Consola** Tu “chat” con R. Donde se ejecutan los comandos y R te devuelve resultados. NO escribas tus comandos en la consola, pues lo que se hace en la consola se pierde. Piensalo como una hoja para hacer un borrador antes de un examen de matemáticas.

**Editor:** Tu bloc de notas donde escribes tus comandos y los guardas para ejecutarlos después. *OJO:* Siempre escribe tus comandos en tu script (con algunas excepciones).

# ¿Qué es cada cosa? ¿Por qué tengo cuatro pantallas?

**Environment:** Mientras usas R, vas a crear muchas cosas. Todo lo que creas en R (Objetos, vectores, bases de datos, etc.) Se vera en tu environment. Es como un estante.

**Archivos:** Usualmente trabajaremos con ptoyectos. Un proyecto es una carpeta que crea R en la que guardas tu código, los archivos que crees para trabajar fuera de R y los archivos que necesites para trabajar dentro de R. **Ver pestañas**

Ya dentro del Script, empezaremos con los conceptos básicos de R.

Imagina que estás construyendo una casa. Necesitas de muchos materiales para poder construir la casa que deseas. Necesitas herramientas. Necesitas crear cosas para tu casa.

R base (Es decir, el R que ya tiene instalado tu computadora) ya te proporciona varias herramientas básicas. A estas herramientas básicas les llamaremos *Comandos*. R base trae todas las herramientas tradicionales que hay en una casa: martillo, desarmador, taladro, etc.

# ¿Qué herramientas me da R base?

## Aritmética básica

```
2+1
```

```
[1] 3
```

```
2-1
```

```
[1] 1
```

```
2*2
```

```
[1] 4
```



# ¿Qué herramientas me da R base?

## Aritmética básica

```
2/2
```

```
[1] 1
```

```
2^2
```

```
[1] 4
```

```
sqrt(4)
```

```
[1] 2
```

# ¿Qué herramientas me da R base?

También me da operadores lógicos. ¿Es verdad o es falso?

```
1 == 1
```

```
[1] TRUE
```

```
2 == 1
```

```
[1] FALSE
```

```
2 != 1
```

```
[1] TRUE
```

```
2 <= 1
```

```
[1] FALSE
```

```
2 < 2
```

```
[1] FALSE
```

```
2 < 10
```

```
[1] TRUE
```

```
2 > 3
```

```
[1] FALSE
```

```
2 > 1
```

```
[1] TRUE
```

```
2 <= 1
```

```
[1] FALSE
```

```
2 <= 2
```

```
[1] TRUE
```

```
3 >= 4
```

```
[1] FALSE
```

```
3 >= 1
```

```
[1] TRUE
```

# Objetos

La herramienta principal de R: **Los Objetos**

Un objeto es la “data structure” fundamental de r: es un texto que puede contener lo que sea.

Piensenlo como un tupper donde puedes guardar muchas cosas

El objeto se crea con esta lógica:

`nombre_del_objeto <- cosa`, o bien `nombre_del_objeto = cosa`

Por ejemplo, texto:

```
perrito <- "firulais"
```

```
perrito
```

```
[1] "firulais"
```

¿Cambió algo en su pantalla? ¿Qué cambió? Recuerden siempre correr sus líneas de código con `command` + `enter`.

Los objetos no se limitan a guardar una cosa, pueden guardar varias al mismo tiempo

```
gatos = c("morris", "dante")  
gatos[c(2)]
```

```
[1] "dante"
```

y también, números.

```
five <- c(1, 2, 3, 4, 5)  
one_to_five <- c(1:5)
```

Podemos sumar objetos

```
estudiantes_pap <- 7  
estudiantes_ri <- 9  
estudiantes <- estudiantes_ri + estudiantes_pap  
estudiantes
```

```
[1] 16
```



Estas y otras herramientas están incluidas en la caja nos da R base. pero, ¿Qué pasa si necesito cortar madera para construir una mesa y no tengo los materiales necesarios?

Así como en la vida real, en R podemos pedir nuevas “cajas” que nos dan herramientas nuevas. Ellas se llaman **librerías**.

Pero primero las tenemos que “Comprar” o instalar en nuestro equipo

```
#install.packages(fivethirtyeight)
```

```
library(fivethirtyeight)
```

Instalamos “fivethirtyeight”, que es una librería que nos da acceso a los datos publicados por la empresa. OJO: las herramientas de la librería no funcionan si no llamas a la librería antes.

Para más info, vean `help(fivethirtyeight)` o vean la pestaña de help.

Así como un objeto puede contener palabras, números o niveles, puede tener bases de datos.

```
trump <- trumpworld_issues
```

```
View(trump)
```

“trump” es un objeto que contiene una base de datos de 185 observaciones de 6 variables.

# Exploremos nuestra base de datos

```
head(trump, n = 3) #Las primeras tres observaciones de la BDD
```

	country	net_approval	approve	disapprove	dk_refused	issue
1	Canada	-64	15	79	5	1
2	France	-81	9	90	1	1
3	Germany	-88	5	93	2	1

```
tail(trump, n = 3) #Las últimas tres observaciones de la BDD
```

	country	net_approval	approve	disapprove	dk_refused	issue
183	Mexico	-66	14	80	6	5
184	Peru	-51	21	72	7	5
185	Venezuela	-54	19	73	8	5

```
names(trump) #Qué variables hay
```

```
[1] "country"          "net_approval" "approve"      "disapprove"  
[6] "issue"
```

```
nrow(trump) #Cuántas observaciones hay
```

```
[1] 185
```

```
ncol(trump) #Cuántas variables hay
```

```
[1] 6
```

También podemos obtener una muy buena descripción con:

```
#install.packager(Hmisc)
library(Hmisc)
Hmisc::describe(trump)
```

trump

6 Variables      185 Observations

-----  
country

n	missing	distinct
185	0	37

lowest :	Argentina	Australia	Brazil	Canada
highest:	Tunisia	Turkey	United Kingdom	Venezuela

-----

net\_approval

n	missing	distinct	Info	Mean	nMedian	Cr
---	---------	----------	------	------	---------	----

¿Qué valores puede tomar cada variable de mi base?

```
unique(trump$country)
```

```
[1] Canada      France      Germany     Greece
[6] Italy       Netherlands Poland       Spain
[11] United Kingdom Russia      Australia   India
[16] Japan       Philippines South Korea  Vietnam
[21] Jordan      Lebanon     Tunisia     Turkey
[26] Kenya     Nigeria     Senegal     South Africa
[31] Argentina   Brazil      Chile       Colombia
[36] Peru        Venezuela

37 Levels: Argentina Australia Brazil Canada Chile Colombia Fr
```



Creemos una nueva variable.

Si en un país la aprobación sobre un issue de trump es mayor a 0, hagamos una variable lógica.

```
trump$liked <- (trump$net_approval > 0)
```

¿Cuántas observaciones hay en las que trump sí sea gustado?

```
library(dplyr)
count(trump, liked)
```

```
# A tibble: 2 x 2
  liked      n
  <lgl> <int>
1 FALSE   181
2 TRUE     4
```