

Curso R / Rstudio 2022

INSTRUCTOR: ALEJANDRO TRUJILLO



Material del curso

Esta presentación, los códigos, archivos, temario, y demás del curso están (o estarán) disponibles en el siguiente github:

<https://github.com/Alextruj/Curso-R-2022>

Descargar archivos:

- Click en archivo > Raw > click derecho “guardar como”



¿Por qué aprender R?

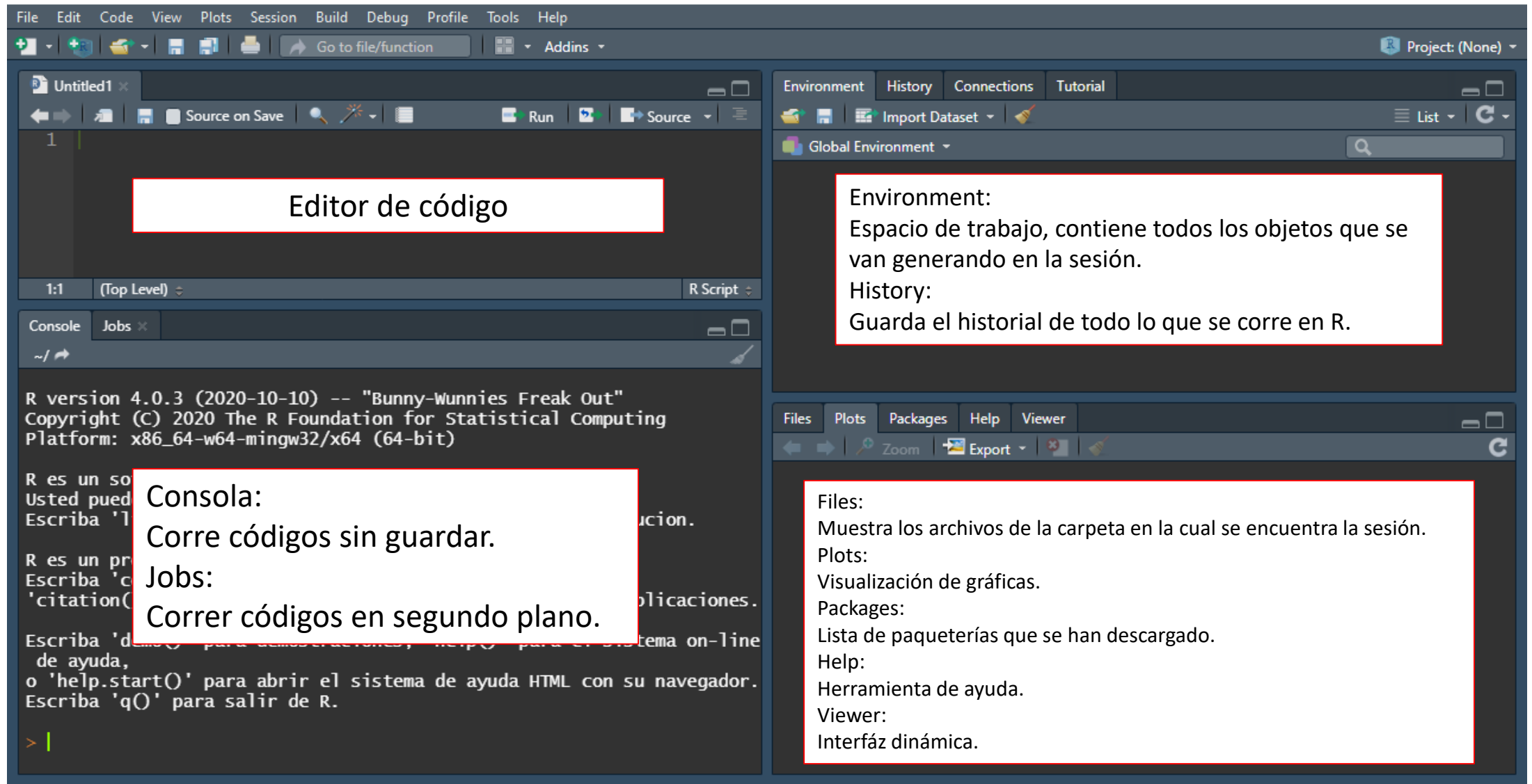
- R es de código abierto (gratis)
- Junto con Python, R es el lenguaje de facto para la ciencia de datos
- Ampla comunidad que incluye economistas y estadistas
- Librerías de visualización robustas



Rstudio es un IDE (entorno de desarrollo integrado) para el lenguaje de programación en R, dedicado a la computación estadística y gráficos (Wikipedia).

¿Por qué usarlo?

- Muy amigable
- No más ventanas
- Código abierto
- Path dependency



Checks de las versiones de R

```
> session.info()
```

```
> RStudio.Version()
```

Parte I: Fundamentales

R como calculadora

Estructura básica de datos

Vectores

Matrices

Data frames

Listas

Funciones básicas

Funciones propias

Loops: for, while, if

R como calculadora

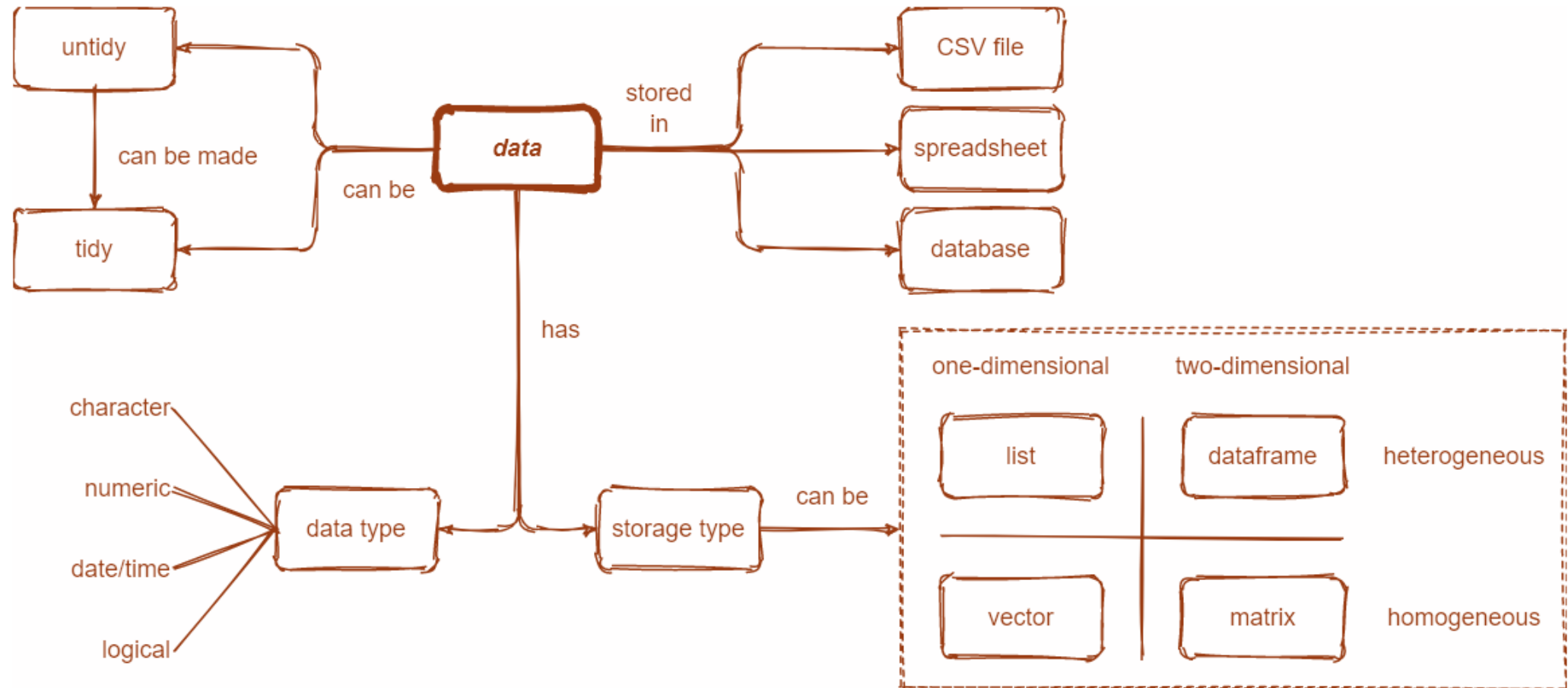
En R puedes hacer las mismas operaciones que en cualquier calculadora, de manera muy intuitiva.

Operación	Símbolo en R
Suma	+
Resta	-
Multiplicación	*
División	/
Potencia	^
...	

Operación	Función en R
Raíz	sqrt()
Exponencial	exp()
Logaritmo	log()
Valor absoluto	abs()
...	

➤ Vamos al código “Calculadora.R”

Estructura básica de datos



Fuente: Computational Thinking for Social Scientists

Estructura básica de datos

- Todo es un objeto
 - Todo tiene un nombre
- } Object Oriented Programming (OOP)



```
> X <- 3
```

```
> X = 3
```

- Vamos al código “Tipos de datos.R”
- Vamos al código “Estructuras basicas.R”

Vectores

La unidad principal de R

- Se crean con la función `c()`

- Propiedades:

- Clase: `class()`

- Tamaño: `length()`

Python: `type()`, `len()`

➤ Vamos al código “Vectores.R”

➤ Vamos al código “Matrices.R”

Data Frames

- Esta es la estructura que más usaremos en el curso
- Matriz heterogénea: permite tener columnas de fechas, números, caracteres, etc.
 - Permite nombrar a las columnas (i.e. tendremos variables)
- Se crean con la función `data.frame()`

```
> paises <- data.frame(Pais = c("Argentina", "Brasil", "Colombia", "Estados Unidos", "Mexico"),  
                        GDP = c(-9.9, -4.1, -6.8, -3.4, -8.2),  
                        Year = c(2020, 2020, 2020, 2020, 2020))
```

- Algunas propiedades:
 - Clase: `class()`
 - Número de variables (columnas): `ncol()`, nombre de variables: `names()` / `colnames()`
 - Número de observaciones (renglones): `nrow()`, nombre de renglones: `rownames()`

➤ Vamos al código “Data frames.R”

Listas

- Una lista es una colección de objetos que puede ser de cualquier clase y cualquier estructura de R
- Nos servirán para guardar data frames o para guardar especificaciones
 - Por default, las gráficas se guardan como lista

```
> Ejemplo_lista <- list("Rojo", "Verde", c(21,32,11), TRUE, 51.23, 119.1, 5+3i)  
  
> Ejemplo_lista
```

Funciones básicas

- En R puedes programar con matrices y vectores las operaciones que quieras realizar, pero lo más común es usar funciones.

- Funciones:
 - Base R
 - Paqueterías
 - Funciones propias

- Acceder a la ayuda de cada función:
 - `help(funcion)`
 - `?funcion`

- Las siguientes dan exactamente lo mismo:

- `mean(x = c(5,2), trim = 0, na.rm = FALSE)`
- `mean(c(5,2), 0, FALSE)`
- `mean(c(5,2))`
- `mean(c(5,2),0)`

➤ Vamos al código “Funciones base.R”

➤ Vamos al código “Funciones base – numeros aleatorios.R”

Qué es cada parámetro

función {de qué paquetería viene}

```
mean {base} R Documentation

Arithmetic Mean

Description
Generic function for the (trimmed) arithmetic mean.

Usage
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)

Arguments
x      An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex
      vectors are allowed for trim = 0, only.
trim   the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that
      range are taken as the nearest endpoint.
na.rm  a logical value indicating whether NA values should be stripped before the computation proceeds.
...    further arguments passed to or from other methods.
```

Opciones default de cada argumento

Paqueterías

- Una paquetería es un conjunto de funciones programadas por alguien más y hechas públicas para descarga.
- La primera vez que vamos a usar una paquetería hay que decirle a R que la instale
 - `install.packages("paquetería")`
- Todas las paqueterías se instalan en la librería de R. Al inicio de cada sesión hay que pedirle a R que la cargue
 - `library(paquetería)`

```
> library()
```

- Una lista de todas las paqueterías disponibles para descarga: [CRAN Packages By Name \(r-project.org\)](https://cran.r-project.org)
- Vamos al código "Paqueterias.R"

Funciones propias

- En R puedes programar tus propias funciones.
- Las partes para crear tu propia función son las siguientes:
 - Nombre de la función
 - Argumentos (inputs)
 - Cuerpo
 - Valor (output)

```
      Nombre      Argumentos
      └──┬──┘      └──┬──┘
> Presentacion <- function(nombre, edad, pais){
Cuerpo → x <- paste("Hola mi nombre es ", nombre, "tengo ", edad, "años y nací en ", país)
Valor  → x
      }
```

➤ Vamos al código "Funciones propias.R"

If... Else

- La mayoría de las condiciones se escriben de la manera más usual o intuitiva
- La manera más usual es dentro de condiciones if... else

```
> A <- 10
> B <- 15

> if (A > B){
  print("A es mayor que B")
} else if (A < B){
  print("B es mayor que A")
} else{
  print("A es igual a B")
}
```

Condición	Operador en R
Igual a	==
Distinto de	!=
Mayor o igual a	>=
Menor o igual a	<=
Menor a / mayor a	< / >

For loop

- Muchas (muchas) veces es útil iterar
 - Por ejemplo: iterar en las columnas de un data frame o iterar en descarga de bases de datos

```
> for (i in 1:10){  
  print(i)  
}
```

While loop

- En el caso de un while, el loop continúa siempre que una condición sea verdadera (y se detiene a partir de la iteración donde es falsa)

```
> i <- 1  
> while (i <= 10) {  
  print(i)  
  i <- i + 1  
}
```

➤ Vamos al código “Loops.R”