

Laboratorio de Econometría

Clase 01

José Burgos

25/09/2023

- R y Rstudio
- Instalación
- Conceptos fundamentales

Section 1

R y Rstudio

R y sus usuarios

R es un lenguaje de programación y entorno computacional enfocado en estadística. A diferencia de otros lenguajes, R se diseñó exclusivamente para tareas estadísticas. Instalar R en una computadora implica configurar el entorno computacional necesario para escribir y ejecutar instrucciones en código.



Rstudio - un IDE para R

Se puede usar R directamente, pero para trabajar con proyecto es recomendable utilizar Rstudio, que es un entorno integrado de desarrollo (IDE, por su siglas en inglés).

¿Para que se usa R?

Usos de R más comunes

- **Manipulación de datos:** R ofrece herramientas para limpiar, transformar y manipular datos, incluyendo la biblioteca `dplyr`.
- **Visualización interactiva:** Crear visualizaciones interactivas y dashboards con bibliotecas como `Shiny`.
- **Estadísticas descriptivas:** Realizar análisis de estadísticas descriptivas, como cálculos de media, mediana, moda, desviación estándar, etc.

Section 2

Instalación

Funcional en:

- Windows
- macOS
- Linux

Page of Posit

- 1 R: <https://cran.r-project.org/bin/windows/base/>
- 2 Rstudio: <https://posit.co/downloads/>

Section 3

Conceptos fundamentales

Entorno computacional

La consola de R veremos donde es ejecutado, solo nos dará el resultado de lo que pidamos allí mismo. En R, *ejecutar*, *llamar* o *correr* implica dar una instrucción o entrada al programa para que realice una tarea en específica.

Los scripts son archivos de texto con extensión `.R` que contienen código que R puede leer y ejecutar.

Directorio de trabajo

El directorio de trabajo en R es donde se encuentran los archivos con los que se trabaja y es el lugar predeterminado para importar y exportar archivos, a menos que se indique lo contrario. Para consultar tu directorio de trabajo utiliza `getwd()`

Así como en matemáticas, una **constante** es un valor inmutable, mientras que una **variable** es un contenedor que puede cambiar de valor durante la ejecución de un programa.

Ejemplo

$$a = \pi r^2$$

Donde π y 2 son constantes, en cambio a y r son las variables.

Para asignar usaremos `<-`, si queremos asignarle el valor de 10 a `chicle` la forma adecuada es:

```
chicle <- 10  
chicle
```

```
## [1] 10
```

Consideraciones

Crear una variable suele ser muy fácil, pero se debe tener en cuenta que los nombres deben de ser CLAROS, NO AMBIGUOS y DESCRIPTIVOS.

Funciones y documentación

Las funciones son operaciones que requieren argumentos para ejecutarse y su nombre generalmente indica su propósito. Funciones comunes:

- ① `mean()`
- ② `median()`
- ③ `quantile()`
- ④ `sum()`

Documentación

Accede a la información de una función o a un paquete con `?mean` o utilizando `help`.

Paquetes

R se expande con paquetes que son colecciones de funciones diseñadas para tareas específicas. Para instalar un paquete, se usa `install.packages()` y para utilizarlo, se emplea `library()`. Es importante recordar que debes cargar el paquete con `library()` cada vez que inicies una sesión en R, ya que las sesiones comienzan con funciones y objetos básicos.



Section 4

Tipos de datos en R

Tipos de datos

Los tipos de datos más común en R son:

Tipo	Ejemplo	Nombre en inglés
Entero	1	integer
Numérico	1.3	numeric
Cadena de texto	"uno"	character
Factor	uno	factor
Lógico	TRUE	logical
Perdido	NA	NA
Vacio	NULL	null

Para consultar el tipo de datos, usaremos la función `class()`

Podemos manipular los datos en R, o más precisamente, realizar conversiones forzadas de un tipo a otro utilizando las funciones de la familia `as.integer()`, `as.numeric()`, `as.character()`, `as.factor()`, `as.logical()`, `as.null()`. Todas estas funciones trabajan con datos y vectores.

Ejemplos:

```
as.character(5)
```

```
## [1] "5"
```

```
as.logical(1)
```

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

Section 5

Operadores

Operadores aritméticos

Tal como sugiere su nombre, estos operadores se emplea específicamente para realizar cálculos aritméticos. Es factible efectuar estas operaciones utilizando datos tanto enteros como numéricos.

Operador	Operación	Ejemplo	Resultado
+	Suma	$5+3$	8
-	Resta	$5-3$	2
*	Multiplicación	$5*3$	15
/	División	$5/3$	1.666667
^	Potencia	5^3	125
%%	División entera	$5\%3$	2

Operadores relacionados

Se usan los operadores lógicos para hacer comparaciones y su resultado siempre será TRUE o FALSE (verdadero o falso, respectivamente)

Table 3: Casi siempre cuando se hacen comparaciones siempre obtendremos TRUE o FALSE.

Operador	Comparación	Ejemplo	Resultado
<	Menor que	5 < 3	FALSE
<=	Menor o igual que	5 <= 3	FALSE
>	Mayor que	5 > 3	TRUE
>=	Mayor o igual que	5 >= 3	TRUE
==	Exactamente igual que	5 == 3	FALSE
!=	No es igual que	5 != 3	TRUE

Operadores lógicos

Los operadores lógicos son usados para describir relaciones lógicas, expresadas como verdadero o falso.

Operador	Comparación
$x \mid y$	x ó y es verdadero
$x \& y$	x Y y son verdaderos
$!x$	x no es verdadero

Advertencia

- $|$ devuelve TRUE si alguno de los datos es TRUE
- $\&$ solo devuelve TRUE si ambos datos es TRUE
- $|$ solo devuelve FALSE si ambos datos son FALSE
- $\&$ devuelve FALSE si alguno de los datos es FALSE

Operadores de asignación

Los operadores de asignación son los más importantes, nos permiten asignar datos a variables. Se recomienda utilizar `<-` a la hora de asignar.

```
juan <- 5  
jose = 5
```

Orden de operaciones

- 1 ^
- 2 * /
- 3 + -
- 4 < > <= >= == !=

Section 6

Estructuras de datos

Vectores

Un vector es una colección de uno o más datos del mismo tipos. Los vectores siempre tienen tres propiedades:

- 1 **Tipo**
- 2 **Longitud**
- 3 **Atributos**

```
vec <- 1 # Crear un vector  
# Vectores de más de un elemento usar c()  
vector1 <- c(1,3,4,5,6)  
vector2 <- 1:10
```

```
vector2 + 2 # Vectorización
```

```
## [1] 3 4 5 6 7 8 9 10 11 12
```

Data frames

Los data frames son como tablas de datos bidimensionales que pueden contener diferentes tipos de datos en sus celdas, lo que los hace heterogéneos. Son ampliamente utilizados en análisis de datos y son similares a las matrices, pero con la flexibilidad de admitir tipos de datos mixtos en filas diferentes, aunque mantienen la restricción de que cada columna debe contener un solo tipo de dato.

##	Sepal.Length	Sepal.Width	Petal.Length
## 1	5.1	3.5	1.4
## 2	4.9	3.0	1.4
## 3	4.7	3.2	1.3

Section 7

Gráficos

Distribución Normal

