



# CLASE INTRODUCTORIA

Manejo Básico de R





# Breve Historia de R

R es un lenguaje de programación utilizado en el análisis de datos, fue creado en 1991 por Ross Ihaka y Robert Gentleman, para la creación de R, Ross y Robert se basaron en 'S', un lenguaje de programación creado a finales de los 70'. Actualmente, R es uno de los software más utilizados para el análisis de datos ya que otorga una gran libertad a los usuarios al momento de realizar distintos tipos de análisis.



# Operaciones Básicas

Una forma sencilla de familiarizarse con R, es utilizarlo como una calculadora, dónde se pueden realizar operaciones básicas como las siguientes:

Operación	Operador	Ejemplo
Adición	'+'	2+2
Sustracción	'-'	10-5
Multiplicación	'*'	7*4
División	'/'	81/9



# Operaciones Básicas

Potenciación	'^'	3^3
Radicación	sqrt() y '^'	sqrt(9) , 27^(1/3)
Valor Absoluto	abs()	abs(-15)
Exponencial y Logaritmos	exp(), log(x, base)	exp(10), log(10), log(10,10)

En el caso de log(), cuando no se coloca ningún argumento de base, por defecto la función realiza la operación de un logaritmo neperiano, es decir, logaritmo con base exp(1)



# Operaciones Básicas

R también permite realizar combinaciones entre las operaciones básicas como por ejemplo:

$$\frac{\ln(27) + |25 - 9|}{e^5 * \sqrt{9}}$$



# Creación de Variables

Cuando se quieren realizar distintas operaciones pero mantener algún valor sin tener que copiarlo de nuevo, se pueden crear variables que almacenan los resultados que se obtiene de las operaciones, para esto se utiliza una flecha '<-' para indicar que se desea guardar el resultado de la operación del lado derecho de la flecha en el lado izquierdo, de tal manera que al escribir el nombre de la variable, se muestre el resultado de la operación.

*Variable ← Operación*



# Operaciones entre variables

Así como se pueden crear variables, R también permite realizar operaciones entre las mismas, como por ejemplo:

$$A < -5 * \text{sqrt}(225)$$
$$B < - \text{exp}(19) * \text{log}(20)$$
$$A * B$$



# Tipos de Datos

Dato	Características
Numéricos	Permiten realizar operaciones múltiples operaciones que involucren números, estos se pueden dividir en <i>integer</i> , <i>numeric</i> , <i>complex</i> , <i>double</i> , entre otros.
Lógicos	Permiten realizar inecuaciones y otro tipo de operaciones lógicas, el resultado al realizar una operación lógica siempre será <i>FALSE</i> o <i>TRUE</i> , para denotar si la operación realizada es falsa o verdadera.. Ejem. $2 > 5$
Carácter	Son todos aquellos datos tipo texto, al utilizar o crear un dato tipo texto, siempre deben ser colocados entre comillas (" "), para que R lo reconozca como un carácter.



# Vectores

Los vectores son una forma de almacenar datos que permite contener una serie de valores del mismo tipo que se distinguen entre sí por un índice, son el principal componente de R, estos no son solo una estructura de datos, casi toda actividad realizada en R involucra el uso de vectores, para crear un vector se usa la función `c()`

```
c(2, 5, 7, pi) ; c("ULA", "Mérida", " FACES") ; c(TRUE)
```



# Operaciones entre Vectores

Así como se puede hacer operaciones básicas entre elementos y variables, también se puede hacer operaciones numéricas entre vectores, específicamente entre vectores numéricos, para esto se puede crear variables que contengan los vectores y realizar las operaciones entre estos, en el orden que se encuentran los elementos de los vectores.

```
V1 <- c(5, 3 * 9, pi, sqrt(25), exp(100))
```

```
V2 <- c(7, 9 * 25, abs(-10), log(15), 5^9)
```

```
V1 + V2
```



# Homogeneidad de los Vectores

Los vectores tienen una propiedad particular, estos son homogéneos, es decir, solamente consideran un tipo específico de dato al momento de ser creados, si por ejemplo se tiene un vector de éste tipo:

$$Vec <- c(70, "ULA", TRUE)$$

Al momento de escribir el nombre de la variable, se puede observar que todos los elementos del vector son de tipo carácter. Para poder romper ésta regla, existen las *listas*.



# Listas

Las listas permiten romper con la homogeneidad de los vectores al no poder tener distintos tipos de datos en un mismo vector, en una lista se puede tener cualquier tipo de dato, y éste mantendrá su propiedad, para crear una lista, se utiliza la función `list()`.

*Lista < -list(70, "FACES", TRUE)*



# Matrices

Una matriz en R, viene siendo un vector de dos dimensiones, el cual se encuentra ordenado por filas  $i$  y columnas  $j$ , con las matrices se pueden hacer las típicas operaciones de matrices, para poder crear una matriz se utiliza la función `matrix()`

$$Matx <- matrix(data = , nrow = , ncol = , byrow = )$$

- *data*, indica los datos que se quieren utilizar, se puede utilizar la función `sample()` para generar datos aleatorios de la siguiente forma:

$$sample(x, size)$$



# Matrices

Dónde  $x$ , denota el vector numérico de datos del que se quiere obtener la muestra, y  $size$ , el tamaño de la muestra. Ejem

`sample(5:100,20)` ; ésta función quiere decir que se quiere una muestra de números del 5 al 100, el símbolo ':' indica el rango de datos.

- *nrow*: indica el número de filas que se desea tenga la matriz
- *ncol*: el número de columnas que se quiere tenga la matriz
- *byrow*: si se quiere que la matriz sea ordenada por filas hay que colocar TRUE, si se quiere por columnas hay que colocar FALSE



# Operaciones entre Matrices

Se pueden realizar operaciones entre matrices de la misma forma que de manera manual, para esto los operadores a utilizar son:

Operación	Operador
Adición y sustracción	'+' ; '-'
Multiplicación	'%*%'
Transpuesta de una matriz	t()
Inversa de una matriz	solve()



# Ejercicio

Crear una matriz 'A' que contenga números aleatorios del 1 al 100 y de tamaño 20 , que tenga 5 filas y 4 columnas, ordenada por filas, luego crear una matriz 'B' que contenga números aleatorios del -100 al -1 y de tamaño 20, que tenga 4 filas y 4 columnas, ordenado por columnas. Obtener la multiplicación entre la matriz A y la matriz B, calcular la transpuesta de A y la inversa de B.