Funciones y matrices

Herramientas de programación para el análisis de datos

Clase 6

Universidad de los Andes
Facultad de Economía

¿Cómo se sienten hasta ahora?

Calcular el promedio de una secuencia de números

Sentados frente a un computador ¿cómo lo hacen? (en algún lenguaje cualquiera)

- Tomar valores y sumarlos, después dividirlos entre el número de valores
- Invocar la función de promedio y decirle qué valores utilizar

Universidad de los Andes
Facultad de Economía

¿Han utilizado matrices alguna vez?

Hoy

- Funciones
- Matrices
- Ejercicio para prototipar código

Funciones pequeño ejemplo: invocar el promedio

```
// Tenemos una variable, e invocamos descriptivas
sum nombreDeLaVariable
```

```
valores <- c(1, 2, 3, 4)
promedio <- mean(valores)
```

```
import statistics
valores = [1, 2, 3, 4]
promedio = statistics.mean(valores)
```

- Nos sirven para ejecutar tareas específicas.
- Son útiles cuando esas tareas se repiten.
- Para usarlas tenemos que darles información en cada situación (parámetros).
- Para usar una función buscamos en internet la documentación

Un pequeño foco en Stata

valores
334
3245
5643
5683

```
sum valores
// Se muestra en la pantalla la media en una tablita
display r(mean)
// Llamando al escalar mean podemos usar el valor para cualquier cosa
```

CASO 1 (lo común)

Queremos saber y usar la media. Entonces usamos **`sum`**.

Un pequeño foco en Stata

valores	nuevaCol
334	3726.25
3245	3726.25
5643	3726.25
5683	3726.25

¿Qué creen que va a ocurrir? (Un minuto para pensar)

```
egen nuevaCol = mean(valores)
```

CASO 1 (lo común)

Queremos saber y usar la media. Entonces usamos **`sum`**.

CASO 2 (uno avanzado)

Queremos crear una columna nueva con el valor de la media. Entonces usamos **egen**

¡Ejercicio!

```
valores:
5, 7, 9, 2, 4
```

- Escoja un lenguaje cualquiera.
- Arme el siguiente arreglo/columna de valores
- Busque para ese lenguaje una función que le permite **saber el valor máximo**

Universidad de los Andes
Facultad de Economía

¡Ejercicio!

```
Respuesta: 9
```

- Escoja un lenguaje cualquiera.
- Arme el siguiente arreglo/columna de valores
- Busque para ese lenguaje una función que le permite **saber el valor máximo**

¿Cómo crear nuestras propias funciones?

Digamos que quieren calcular un indicador que ustedes se inventaron

Crear nuestras propias funciones

```
// En Stata: normalmente no creamos funciones nuevas, todo queda en el do-file pero se puede hacer agregando archivos ado (nuevos programas) al sistema
```

```
nombre_de_la_fun <- function(arg_1, arg_2, ...) {
   return(3)
}
```

```
def nombre_de_la_fun(arg_1, arg_2, …):
        return 3
```

Hay que definir:

- El nombre con el que se va a invocar
- Los parámetros
- El proceso
- Lo que retorna

¿Cómo sería esta definición para la función media?

Detalle de R

```
// En Stata: normalmente no creamos funciones
n
h
s
```

```
cuentame <- function(valor1){
  salida<-valor1 + 1
  return(salida)
}

# ¿Qué valor retorna?
cuentame(2)
```

Retorna 3

```
d
```

Cuando no especificamos qué valor retornar, R retorna la última expresión evaluada.

Universidad de los Andes
Facultad de Economía

Crear nuestras propias funciones

```
// En Stata: normalmente no creamos funciones nuevas, todo queda en el do-file pero se puede hacer agregando archivos ado (nuevos programas) al sistema
```

```
nombre_de_la_fun <- function(arg_1, arg_2, ...) {
   # Cuerpo de la función
   return(3)
}
```

```
def nombre_de_la_fun(arg_1, arg_2, …):
       return 3
```

Hay que definir:

- El nombre con el que se va a invocar
- Los parámetros
- El proceso
- Lo que retorna

Nombren cada uno en el caso de la función media

Universidad de los Andes
Facultad de Economía

El caso de Python

```
// En Stata: normalmente no creamos funciones
nuevas, todo queda en el do-file pero se puede
hac
sis
```

```
nom



}
```

```python
def unaFuncionMuyBuena(dato1, dato2):
        retornar = dato1+dato2
        return retornar

# ¿Qué valor retorna?
unaFuncionMuyBuena(2,3)
```

En Python, si no especificamos el valor de retorno, entonces retorna **None**

Ejercicio de dificultad intermedia (10 minutos)

En R: `datos <- c(6,3,6,3,2)`
En Python: `datos = [6,3,6,3,2]`

Defina una función en uno de los lenguajes que está utilizando que:

- Tome **un vector de valores**
- Recupere el primer valor
- Recupere el último valor de la lista
- Retorne la suma de esos dos

Ejercicios activos

```
Respuesta: 8
```

La clave es utilizar la función longitud:

```
En R: length(datos)
En Python: len(datos)
```

Veamos un par de ejemplos.

Universidad de los Andes
Facultad de Economía

Matrices

Advertencia: hay dos formas de "arreglar" números

¿Para qué se usan?

Matrices

$$\begin{bmatrix} 2 & 5 & 5 & 1 \\ 2 & 4 & 9 & 3 \\ 2 & 3 & 3 & 1 \\ 6 & 7 & 6 & 8 \end{bmatrix}$$

s: llámense base de datos, dataFrames, etc.

Columna 1	Columna 2	Columna 3	Columna 4
2	5	5	1
2	4	9	3
2	3	3	1

Ustedes han visto ya este un poco y lo exploraremos más en el módulo 4

Detrás de bambalinas

- Las funciones permiten simplificar cálculos con los datos:

una media, la mediana, hacer una regresión…

- Detrás de esos cálculos hay matrices, las funciones tienen a programarse con ellas para hacerlas más eficientes.

Las regresiones lineales son operaciones matriciales

- En otras palabras: son matrices <u>detrás de bambalinas</u>

- Mucha de la estadística se construye usando ***Álgebra lineal***

Universidad de los Andes
Facultad de Economía

Para qué las usamos en el mundo de los datos

- En Stata: a veces **hacemos cálculos y queremos guardar los resultados** para luego exportarlos.
- Podemos **implementar a mano** funciones que ya existen para modificar algo particular sobre su metodología o ajustarlas.
- ¡Cuando **jugamos con matemáticas**! Por ejemplo queremos estudiar cómo se comporta un objeto o un gráfico creado con una fórmula.

Juguemos con matrices

Les presento a A, B y C

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

Suma A + B (en vivo)

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A+B = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Resta A – C (Ejercicio)

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

$$A\text{-}C = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en <u>FILA 1 COLUMNA 2</u>, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & \square & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la <u>FILA 1 de A</u> y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} \\ \\ \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la <u>COLUMNA 2 de B</u> y:

$$AC \begin{bmatrix} \quad & \quad & \quad \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1} & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & \boxed{1} & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{\boxed{1} \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & 1 \cdot 1 + & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad \boxed{2} \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & \boxed{2} & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & 1 \cdot 1 + 2 \cdot 2 + & \\ & & \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 2 & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} 1 + 2 \cdot 2 + 3 \cdot 2 \\ \\ \end{bmatrix}$$

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & 1 + 4 + 3 \cdot 2 & \\ & & \\ & & \end{bmatrix}$$

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} & 1 + 4 + 6 & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} \boxed{} & 11 & \\ & & \\ & & \end{bmatrix}$$

Ejercicio

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} 1 \cdot 2 + 2 \cdot 1 + 3 \cdot 1 & 11 & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación (qué onda) AC

$$A = \begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

FILA, COLUMNA: La posicion en FILA 1 COLUMNA 2, es tomar la FILA 1 de A y la COLUMNA 2 de B y:

$$AC \begin{bmatrix} 7 & 11 & \\ & & \\ & & \end{bmatrix}$$

Universidad de los Andes Facultad de Economía

Nota

A por B

no es igual a

B por A

En matrices, el orden importa

Universidad de los Andes
Facultad de Economía

Una matriz útil: la matriz transpuesta

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Cada filas se vuelve una columna:

$$A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Transponer:

$$C = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}$$

$$C^T = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 3 & 1 \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Multiplicación de matriz por escalar

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 2 \cdot B = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 0 & 2 \cdot 0 \\ 2 \cdot 0 & 2 \cdot 1 & 2 \cdot 0 \\ 2 \cdot 0 & 2 \cdot 0 & 2 \cdot 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Universidad de los Andes
Facultad de Economía

Ahora sí matrices en código

Ya sabiendo esto es muy directo

¿Recuerdan los arreglos?

```
// Por ejemplo:
matrix define datos=(6,7,8,9)
display datos[1,2]
```

```
datos <- c(6,7,8,9)
datos[2] # nos muestra 7
```

```
datos = [6,7,8,9]
datos[2] # nos muestra 8
```

Para recuperar un valor de un arreglo, tenemos que comunicar **un dato que indique la posición** del valor que queremos recuperar.

En Stata solo se manejan arreglos matriciales, con filas y columnas.

¿Recuerdan los arreglos?

```
matrix define datos=(6,7,8,9)
display datos[1,2]
```

```
datos <- c(6,7,8,9)
datos[2] # nos muestra 7
```

```
datos = [6,7,8,9]
datos[2] # nos muestra 8
```

Para tomar un valor de una matriz tenemos que indicar **dos datos**: fila y columna

$$\begin{pmatrix} 2 & 4 & 6 \\ 1 & 7 & 3 \\ 0 & 3 & 6 \end{pmatrix}$$

Matrices

```
matrix A = (2,4,6\1,7,3\0,3,6)
```

```
arreglo1 <- c(2,4,6)
arreglo2 <- c(1,7,3)
arreglo3 <- c(0,3,6)
matrix <- cbind(arreglo1, arreglo2, arreglo3)
```

```
A = [[2, 4, 6],
     [1, 7, 3],
     [0, 3, 6]]
```

Para crear esta matriz en cada lenguaje, también las definimos en términos de filas y columnas

Note:
R pega columnas
Python y STATA pegan filas

$$\begin{matrix} 4 & 6 \\ 7 & 3 \\ 3 & 6 \end{matrix}\Bigg)$$

Creando matrices en Stata desde variables

var1	var2
1	8
2	7
3	6
4	5

```
. matrix list A

A[4,2]
     var1  var2
r1      1     8
r2      2     7
r3      3     6
r4      4     5
```

```
mkmat var1 var2, matrix(nombreDeMatriz)
```

En Stata también podemos partir del editor y convertir columnas enteras en matriz

a
a
a
m

A

```
    [0, 3, 6]]
```

Matrices

```
matrix A = (2,4,6\1,7,3\0,3,6)
```

```
arreglo1 <- c(2,4,6)
arreglo2 <- c(1,7,3)
arreglo3 <- c(0,3,6)
matrix <- cbind(arreglo1, arreglo2, arreglo3)
```

```
A = [[2, 4, 6],
     [1, 7, 3],
     [0, 3, 6]]
```

Para crear esta matriz en cada lenguaje, las definimos en términos de filas y columnas

$$\begin{pmatrix} 2 & 4 & 6 \\ 1 & 7 & 3 \\ 0 & 3 & 6 \end{pmatrix}$$

Otras formas de crear matrices en R

```
A <- matrix(1:9, nrow = 3, ncol = 3)
```

```
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
A
   [0, 3, 6]]
```

Hay atajos para crear matrices en R cuando son especiales.

- Números secuenciales
- Ceros
- Identidad

En STATA y Python también

Universidad de los Andes
Facultad de Economía

Matrices

```
matrix A = (2,4,6\1,7,3\0,3,6)
```

```
arreglo1 <- c(2,4,6)
arreglo2 <- c(1,7,3)
arreglo3 <- c(0,3,6)
matrix <- cbind(arreglo1, arreglo2, arreglo3)
```

```
A = [[2, 4, 6],
    [1, 7, 3],
    [0, 3, 6]]
```

Para crear esta matriz en cada lenguaje, las definimos en términos de filas y columnas

$$\begin{pmatrix} 2 & 4 & 6 \\ 1 & 7 & 3 \\ 0 & 3 & 6 \end{pmatrix}$$

En Python también se pueden crear matrices con numPy

```
mat
```

```
arr
arr
arr
mat
```

```
# Importamos numpy
import numpy as np

A = np.matrix('2 4 6; 1 7 3; 0 3 6')
```

Para estadística es más común usar matrices de la librería **numpy**

En Python también se pueden crear matrices con numPy

```
# Definida nativamente
Anat = [[2, 4, 6],
    [1, 7, 3],
    [0, 3, 6]]

# Definida mediante numpy
import numpy as np
Anum = np.matrix('2 4 6; 1 7 3; 0 3 6')
```

```
[40]: Anat

[40]: [[2, 4, 6], [1, 7, 3], [0, 3, 6]]

[41]: Anum

[41]: matrix([[2, 4, 6],
              [1, 7, 3],
              [0, 3, 6]])
```

Para estadística es más común usar matrices de la librería **numpy**

Universidad de los Andes
Facultad de Economía

En Python también se pueden crear matrices con numPy

mat

arr
arr
arr
mat

```python
# Definida nativamente
Anat = [[2, 4, 6],
    [1, 7, 3],
    [0, 3, 6]]

# Definida mediante numpy
import numpy as np
Anum = np.matrix('2 4 6; 1 7 3; 0 3 6')

# elemento en fila 0 columna 1 (OJO CONTANDO DE 0)
```

```
[38]: Anat[0][1]
[38]: 4
[29]: Anum.item((0, 1))
[29]: 4
```

Para estadística es más común usar matrices de la librería **numpy**

No importa si usan matrices nativas o de numPy, pero sean consistentes.

Universidad de los Andes
Facultad de Economía

Tomar elementos de una matriz en R

```
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> A[1,2]
[1] 4
```

A

[0, 3, 6]]

Para seleccionar elementos de una matriz en R se usa la notación con corchetes

Tomar elementos de una matriz en STATA

```
. matrix list A

A[4,2]
    var1  var2
r1     1     8
r2     2     7
r3     3     6
r4     4     5
```

```
. matrix C = A[1,1]

. matrix list C

symmetric C[1,1]
    c1
r1   1
```

```
a
a
a
m
```

```
A
    [0, 3, 6]]
```

Se pueden asignar partes específicas de una matriz: Recuerde que STATA cuenta desde 1.

Operaciones

Suma

```
matrix C = A + B
```

```
C <- A + B
```

```
C = A + B
```

Universidad de los Andes
Facultad de Economía

Resta

```
matrix C = A - B
```

```
C <- A - B
```

```
C = A - B
```

Universidad de los Andes
Facultad de Economía

Multiplicación

```
matrix C = A*B
```

```
C <- A%*%B
```

```
C = A*B
```

En R consideran que la multiplicación es más común es elemento a elemento, por eso la notación simple está reservada para "elemento por elemento"

Multiplicación elemento por elemento

```
// No acepta (hay que programarla con loops)
```

```
C <- A*B
```

```
C = np.multiply(A,B)
```

En python las librerías para matrices están programadas en numpy.

Transposición: filas a columnas

```
matrix C = A'
```

```
C <- t(A)
```

```
np.transpose(A)
```

Aquí sí, la notación de cada lenguaje es única y especial.

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

Universidad de los Andes
Facultad de Economía

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

Universidad de los Andes
Facultad de Economía

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

Ejercicio agrupador: prototipando código

Vamos a crear una función en R o Python que tome una matriz:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Y la vamos a cambiar componente a componente sumándole uno.

- En la página está esta presentación para referencia
- Vamos a dar 2 minutos para cada prototipo parcial
- Alfredo va a ejecutar el prototipo parcial después: nadie se nos queda

Ejercicio agrupador: prototipando código

1. Dibujar el paso a paso
2. Describir el paso a paso en voz alta (como lo haría un computador)

Ejercicio agrupador: prototipando código

1. Dibujar el paso a paso
2. Describir el paso a paso en voz alta
3. Ustedes: Crear una matriz 3x3 llena de ceros
4. Ustedes: Crear una función que muestre cualquier cosa en pantalla y reciba un parámetro
5. Ustedes: hacer un loop que muestre en pantalla cada número de fila
6. Ustedes: hacer un loop dentro de ese loop que muestre cada número de columna, y que muestre en qué fila va y en qué columna
7. Ustedes: afuera de todo eso, cambiarle el valor a una celda de la matriz (cualquiera)
8. Ustedes: meter dentro del doble loop la asignación anterior