

Ejercicios Básicos: Python con NumPy y Matemáticas

Maestría en Investigación en Ciencia de Datos — FCFM, BUAP

Ejercicio 1: Operaciones Básicas con Arreglos

Crea dos arreglos de NumPy **a** y **b** con los valores indicados y calcula lo siguiente sin usar ciclos for:

- a) La suma elemento a elemento de **a** y **b**.
- b) El producto elemento a elemento de **a** y **b**.
- c) El cuadrado de cada elemento de **a**.
- d) La raíz cuadrada de cada elemento de **b**.
- e) La suma total de todos los elementos de **a**.

```
a = [1, 4, 9, 16, 25]
```

```
b = [2, 4, 6, 8, 10]
```

Funciones sugeridas: np.array(), np.sqrt(), np.sum()

Ejercicio 2: Estadísticas Básicas de un Conjunto de Datos

Dado el siguiente arreglo de calificaciones de un grupo de estudiantes, calcula usando NumPy:

- a) El promedio (media aritmética).
- b) La mediana.
- c) La desviación estándar.
- d) La calificación máxima y la mínima.
- e) Cuántos estudiantes obtuvieron calificación mayor o igual a 70.

```
calificaciones = [55, 78, 82, 90, 61, 74, 88, 95, 70, 63, 77, 85]
```

Funciones sugeridas: np.mean(), np.median(), np.std(), np.max(), np.min(), np.sum()

Ejercicio 3: Operaciones con Matrices

Crea las matrices **A** y **B** con los valores indicados y realiza las siguientes operaciones:

- a) Suma de matrices: $A + B$.
- b) Multiplicación matricial: $A \times B$.
- c) Transpuesta de **A**.
- d) El determinante de **A**.
- e) La traza de **A** (suma de la diagonal principal).

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

```
B = [[9, 8, 7],
```

```
[6, 5, 4],  
[3, 2, 1]]
```

Funciones sugeridas: np.array(), np.dot() o @, .T, np.linalg.det(), np.trace()

Ejercicio 4: Generación de Secuencias y Funciones Matemáticas

Usando NumPy para generar arreglos de valores y evaluar funciones matemáticas:

- a) Genera un arreglo de 100 valores equiespaciados entre 0 y 2π .
- b) Evalúa $f(x) = \sin(x)$ y $g(x) = \cos(x)$ en esos valores.
- c) Verifica la identidad trigonométrica fundamental elemento a elemento:

$$\sin^2(x) + \cos^2(x) = 1$$

- d) Calcula el valor máximo de $|f(x) - 1|$ para comprobar que la identidad se cumple con precisión numérica (el resultado debe ser cercano a 0).

```
# Pista para el inciso c):  
resultado = np.sin(x)**2 + np.cos(x)**2  
# Verifica usando np.allclose(resultado, np.ones(100))
```

Funciones sugeridas: np.linspace(), np.sin(), np.cos(), np.abs(), np.max(), np.allclose()

Ejercicio 5: Conversión de Temperatura y Filtrado de Datos

Tienes un arreglo con temperaturas registradas en grados Celsius durante un mes (30 valores). Usando NumPy:

- a) Convierte todas las temperaturas a grados Fahrenheit usando la fórmula:

$$F = (C \times 9/5) + 32$$

- b) Encuentra cuántos días la temperatura superó los 30°C .
- c) Calcula el promedio solo de los días con temperatura mayor a 25°C .
- d) Devuelve los índices de los 3 días más calurosos.

```
temperaturas_C = [22, 28, 31, 35, 29, 27, 24, 30, 33, 26,  
                  21, 25, 32, 34, 28, 23, 29, 31, 27, 36,  
                  30, 24, 22, 28, 33, 35, 29, 26, 31, 25]
```

Funciones sugeridas: np.array(), np.sum(), np.mean(), np.argsort()

Nota: La funcionalidad del código vale 70% y la calidad del código 30%.