

## Computación, Práctica 10.

### Programación en Python: Numpy y Matplotlib.

Los ejercicios en Python se pueden realizar en las computadoras del aula informatizada de la misma forma que venimos haciéndolo con los ejercicios en FORTRAN, es decir, escribiendo el código fuente en un archivo, en estos casos con extensión **.py**, y luego ejecutándolo por línea de comando en la terminal escribiendo:

```
> python3 archivo.py
```

También se puede utilizar el entorno *Spyder* que se encuentra instalado en las computadoras del aula informatizada. Para acceder al mismo, en la terminal escribimos:

```
> spyder &
```

Este entorno posee su propio editor y su consola de ejecución. Una de las ventajas de *Spyder* es que nos permite ejecutar porciones del programa independientemente del resto.

#### Numpy

Recordar que para usar *Numpy* debe cargar la librería de la siguiente forma (antes de usarla).

```
import numpy as np
```

Recordar que de aquí en adelante toda acción sobre numpy será de la forma:

```
np.algo()
```

**Ejercicio 1.** Crear un objeto ndarray unidimensional con una longitud de 10 y todos los elementos con valor inicial 0, y luego hacer que el quinto elemento sea igual a 1.

**Ejercicio 2.** Crear un objeto ndarray cuyos elementos tomen valores del número 10 al 49.

**Ejercicio 3.** Crear un vector fila  $x$  de 5 elementos equiespaciados cuyo valor se encuentre entre 2 y 3. Usar la función `np.linspace`.

**Ejercicio 4.** Crear dos vectores de 10 elementos. El primero con valores de 20 a 30 y el segundo de -5 a 5. Obtener la raíz cuadrada del primero y el valor absoluto del segundo. Además obtener el máximo y el mínimo de ambos arreglos cuando se comparan elemento a elemento. Utilice las funciones `sqrt`, `abs`, `maximum` y `minimum`.

**Ejercicio 5.** Construir una matriz 3x3 con valores de 0 a 8. Recuerde el uso de la orden `reshape()` para facilitar la operación. Obtener las siguientes propiedades del arreglo: forma (`shape`), tamaño (`size`), y tipo de datos (`dtype`).

**Ejercicio 6.** Construir una matriz identidad de 6x6.

**Ejercicio 7.** Construir un arreglo con valores al azar con forma 3x3x3.

**Ejercicio 8.** Construir una matriz de 10x10 con 1 en los bordes y 0 en el interior utilizando una orden compacta modificando los rangos de índices.

**Ejercicio 9.** Construir dos arreglos matriciales numpy de 10x10. Llenarlos con números reales al azar y multiplicarlos.

**Ayuda:** La función `np.random.rand(n,m)` construye matrices con números al azar donde  $n$  y  $m$  son las dimensiones y la función `np.matmul(a,b)` multiplica las matrices  $a$  y  $b$  en ese orden.

**Ejercicio 10.** Usando la función `random.randint` cree un vector con 10 elementos enteros generados al azar entre 1 y 100. Ordene los elementos de mayor a menor mediante la función `sort` de numpy y muestre el vector resultante en pantalla.

**Ejercicio 11.** Generar dos matrices  $A$  y  $B$  de  $4 \times 2$ , donde los elementos de la primera tengan los valores (por filas) de 0 a 7, y los de la segunda de 5 a 12. Verificar si ambas matrices son iguales. Obviamente las dos matrices serán distintas (si se desea ver que sucede cuando las matrices son iguales hacer  $B = A$ , antes de verificar). Con las matrices originales imprimir el elemento que es igual en ambas matrices. En este caso debería suceder que no hay ningún mismo elemento  $i,j$ -ésimo con igual valor en cada matriz. Cambiar el último elemento de la matriz  $B$  por 7 (de forma que tenga el mismo valor que en la matriz  $A$ ) y volver a ejecutar el programa.

**Ayuda:** Construir una máscara y pedir el elemento `True` de alguna de las matrices.

#### Matplotlib

**Ejercicio 12.** Construir un programa que grafique la función  $f(x) = \sin(x)$  en el intervalo  $[0, 720]$  (cuyos valores están en grados) con una línea continua. Ponerle nombre a los ejes. Recuerde que debe cargar la librería Matplotlib con el siguiente comando:

```
import matplotlib.pyplot as plt
```

**Ayuda:** Si utiliza la librería numpy ahorra muchas líneas de código. Para hacer la línea continua suave, mapee la función cada 10 grados.

**Ejercicio 13.** Rehacer el dibujo anterior pero graficando al mismo tiempo la función  $g(x) = \cos(x)$ . Poner etiquetas a cada dibujo según su color.

**Ejercicio 14.** Rehacer el diagrama anterior pero ahora no usar líneas continuas sino sólo puntos.

**Ejercicio 15.** Rehacer nuevamente el dibujo anterior, pero con línea continua y puntos. Para el  $\sin(x)$ , que los puntos sean rojos y para el  $\cos(x)$  que sean verdes. Agregar un título y una grilla al dibujo.

**Ejercicio 16.** Con los siguientes datos, construir un diagrama de barras, que tenga un título y leyendas referidas a los colores.

```
x1 = [0.25, 1.25, 2.25, 3.25, 4.25]
y1 = [2, 15, 55, 35, 12]
x2 = [0.75, 1.75, 2.75, 3.75, 4.75]
y2 = [24, 42, 15, 39, 26]
```

**Ayuda:** utilice `width = 0.5` para que no se superpongan las barras.

**Ejercicio 17.** En el archivo *Temp.dat* se encuentran tabuladas para los 12 meses del año, las temperaturas máximas y mínimas promedio en la ciudad de La Plata durante 2023.

Leyendo los datos de este archivo, realizar un programa que grafique a lo largo del año los valores promedios de las temperaturas máximas y mínimas. El gráfico debe tener un encabezado, bien identificado que es cada eje, e indicado en las referencias que es cada curva.

**Ejercicio 18.** Durante 1948 se jugó un torneo entre cinco ajedrecistas para dirimir quién sería el nuevo Campeón Mundial, que estaba vacante tras la muerte de Alexander Alekhine.

En la lista *ronda*, se encuentran listada el número de ronda. En las siguientes, está el puntaje acumulado en cada una de las rondas por los jugadores

Mikhail Botvinnik, Vasili Smyslov, Paul Keres, Samuel Reshevsky y Max Euwe, respectivamente.

```
ronda = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
botvinnik = [ 0.0, 1.0, 1.5, 2.5, 3.5, 3.5, 4.0, 4.5, 5.0, 6.0, 6.0, 7.0, 8.0, 8.0, 9.0, 9.0, 9.5, 10.0, 11.0, 12.0, 12.0, 12.5, 13.0, 14.0, 14.0]
smyslov = [0.46, 0.46, 1.0, 1.96, 1.96, 2.46, 2.46, 3.0, 3.96, 3.96, 4.96, 5.46, 5.46, 5.45, 5.46, 6.0, 7.0, 7.5, 8.5, 8.54, 9.04, 9.5, 10.0, 11.0, 11.0]
keres = [ 1.0, 2.04, 2.04, 2.04, 2.04, 2.54, 3.5, 4.04, 4.04, 4.04, 5.04, 5.54, 6.5, 6.5, 6.5, 7.54, 7.54, 7.54, 7.54, 7.54, 8.5, 9.04, 9.54, 9.54, 10.54]
reshevsky = [ 0.54, 0.54, 1.5, 1.5, 2.5, 3.0, 3.0, 3.5, 4.0, 4.5, 4.5, 4.5, 4.5, 5.5, 6.0, 6.5, 6.5, 7.46, 7.46, 8.46, 8.96, 8.96, 9.46, 9.46, 10.46]
euwe = [ 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0, 1.5, 1.5, 1.5, 1.5, 1.5, 2.5, 3.0, 3.0, 3.5, 3.5, 3.5, 3.5, 4.0, 4.0, 4.0, 4.0]
```

Con estos datos, realizar un gráfico que reproduzca al de la figura 1.

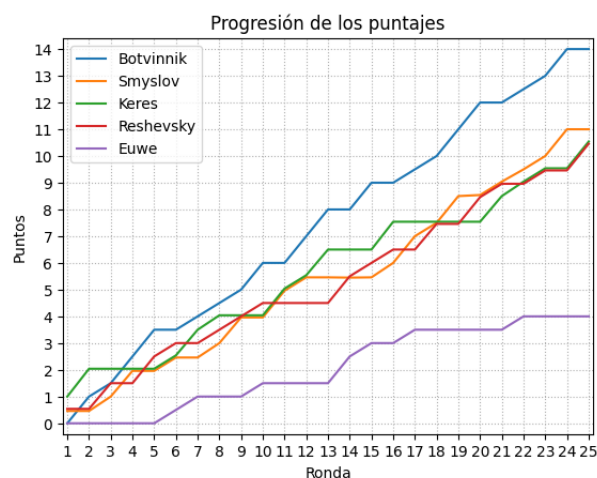


Figura 1