

## 1<sup>ra</sup> Actividad: Listas & Arreglos


A partir de la adjunta lista  $L$  escriba un programa que realice lo pedido a continuación:

```
1 L=[ [1, 200, 50000],  
2 [1, 2, 3],  
3 [1],  
4 [100],  
5 [2, 2, 3, 4, 5, 6, 3, 21],  
6 [1, 2, 5, 2, 5, 7],  
7 [1, 2, 0, 8, 7, 6],  
8 [5, 6, 4, 8, 3, 2],  
9 [2, 3],  
10 [1, 2, 99]]
```

- Calcule el promedio de cada lista contenida en  $L$  y cree una nueva lista que contenga cada uno de los promedios.
- Ordene los valores de menor a mayor a través del método Bubble Sort.
- Transforme la lista a un arreglo NumPy.
- Utilizando funciones de NumPy, calcule el promedio del arreglo generado en el punto anterior y ordene de mayor a menor.

## 2<sup>da</sup> Actividad: Divisores Comunes por Michelle Hamasaki

Defina una función que reciba un número  $n$  y devuelva una lista con los divisores de este. Por ejemplo:  $n = 8 \rightarrow [1, 2, 4, 8]$ . Obtenga dicha lista para  $n = 25$  y  $n = 100$ , luego compare ambos resultados y en una tercera lista, almacene los divisores comunes (Aquellos que se encuentran en las listas de ambos  $n$ ).

 **Hint:** Recordar que  $a\%b$  en python calcula el resto de la división de  $a$  en  $b$ .

### 3<sup>ra</sup> Actividad: Asientos Libres por Michelle Hamasaki

Una sala de cine tiene la ocupación de sus asientos almacenados en un archivo `asientos.dat`, en el cual los asientos vacíos están designados con 0 y los ocupados con 1. Obtenga los datos de este archivo, asígneles a un arreglo y almacene en una lista la posición de los asientos libres.

Verifique gráficamente que la disposición de los asientos es correcta, a través de la herramienta `imshow` de la librería `matplotlib`.

Finalmente, requiera al usuario que ingrese la posición de un puesto libre, para que de esta manera, el programa modifique el arreglo y que el asiento seleccionado pase a estar ocupado, es decir, que cambie de 0 a 1. Nuevamente puede verificarlo gráficamente.

### 4<sup>ta</sup> Actividad: Manchas Solares por Marcela Lopez

Las manchas solares son un fenómeno que aparece temporalmente en la superficie del Sol debido a concentraciones del flujo del campo magnético. En el archivo `sunspots.txt` se le entrega un arreglo que posee la cantidad de manchas solares detectadas por año.

- Escriba un código que determine cual fue el año con mayor avistamiento de manchas solares, a su vez, también indique el año con menos manchas solares observadas
- Escriba un código que separe los datos por décadas, luego que calcule el promedio de manchas solares por década y determine en qué época hubieron mayor cantidad de manchas solares.
- Realice una gráfica de dispersión con los datos inicialmente entregados.

### 5<sup>ta</sup> Funciones, arreglos y gráficos 3D por Gustavo Wörner

La finalidad de esta actividad es graficar la suma de dos funciones, para ello se siguen los siguientes pasos:

- Cree un fichero llamado **grafico3D.py**.
- En este fichero, defina las funciones:

$$f(x, y) = \sin x + \cos y \quad (1)$$

$$g(x, y) = x^2 \cdot y^3 \quad (2)$$

- Luego, cree dos arreglos (X e Y) que contengan 2000 números entre -10 y 10 (*hint*: puede utilizar **np.linspace**).
- Cree un tercer arreglo Z de 2000 x 2000 que contenga sólo ceros.
- Iterando entre los valores de X e Y, almacene el resultado de  $f(x, y) + g(x, y)$  en Z.

- Para graficar en 3D los valores almacenados por el arreglo Z, es necesario importar **mplot3d** de **mpl\_toolkits** (paquete de herramientas de `matplotlib`). Además, se debe crear arreglos del mismo tamaño de Z que contengan los valores de X e Y para cada elemento de Z, lo cual se puede realizar utilizando **np.meshgrid**, tal que:

```

1 import matplotlib.pyplot as plt
2 from mpl_toolkits import mplot3d
3 import numpy as np
4
5 # Ingrese su código aquí
6
7 X, Y = np.meshgrid(X, Y)
8 ax = plt.axes(projection="3d")
9
10 ax.plot_surface(X, Y, Z)
11 plt.show()

```

## 6<sup>ta</sup> Funciones y aleatoriedad por Gustavo Wörner

En esta actividad con el uso de la función  $g(x, y)$  del ejercicio anterior, calcule el área bajo la curva (*Integral*) entre 0 y 1 utilizando números aleatorios. Para esto, considere los siguientes pasos:

- Usando la función **random()** del paquete **random**, cree tres variables: X, Y y Z, cada una con valores aleatorios entre 0 y 1.
- Diseñe un contador (*count*) igual a 0.
- Si el valor de  $g(x, y)$  es mayor a la variable Z, se le suma 1 a *count*.
- Repetir esto 100.000 veces y luego imprimir el valor de:

$$1 \times \frac{\text{count}}{100000} \quad (3)$$

- Finalmente, compare su resultado con el valor de la integral analítica:

$$\int_0^1 g(x, y) \cdot dx dy = \int_0^1 x^2 \cdot dx \cdot \int_0^1 y^3 \cdot dy = \frac{x^3 \cdot y^4}{12} \Big|_0^1 = \frac{1}{12} \quad (4)$$

El método utilizado para calcular integrales en este problema es llamado **Método de Monte Carlo** y es ampliamente utilizado para realizar simulaciones o estimaciones de eventos desconocidos.

## 7<sup>ma</sup> Actividad: Aproximadamente $\pi$

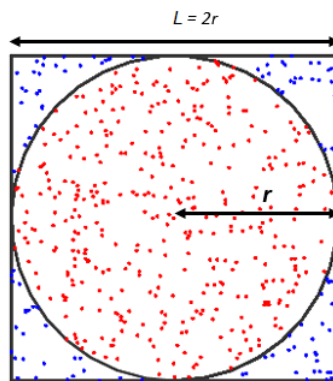
Una forma de estimar el valor de  $\pi$  es usando un método de Monte Carlo. Este consiste en dibujar un cuadrado de lado  $L$  con un círculo interior, generando luego una gran cantidad de puntos aleatorios dentro del cuadrado, para así contar cuántos se encuentran dentro del círculo. De esta manera, para poder obtener un valor aproximado de  $\pi$  se consideran las siguientes afirmaciones:

- El área del círculo es  $\pi r^2$  y la del cuadrado es  $L^2 = (2r)^2 = 4r^2$ .
- Si se divide el área del círculo por el área del cuadrado se obtiene  $\pi/4$ .
- Se puede usar la misma relación entre el número de puntos dentro del cuadrado y el número de puntos dentro del círculo, permitiendo utilizar la siguiente fórmula para estimar el valor de  $\pi$ :

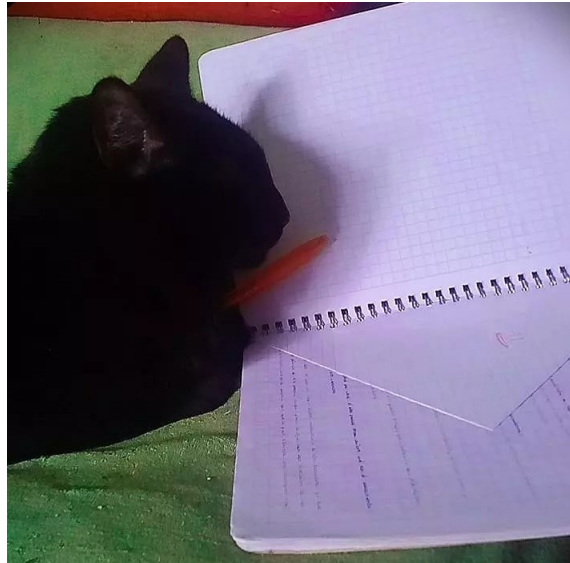
$$\pi \approx 4 \times \frac{\text{Número de puntos dentro del círculo}}{\text{Número total de puntos}}$$

Con toda la información entregada, elabore un programa en `Python` que a través de una función que reciba como parámetro la cantidad de puntos a generar, sea capaz de establecer una aproximación de  $\pi$ . Para realizar esta actividad, considere los siguientes pasos:

- Genere un cuadrado con puntos al azar entre 0 y 1 dentro de éste.
- Cuente a través de un ciclo `for` cuantos puntos se encuentran dentro del círculo. (**Debe realizar un gráfico de scattering que muestre la figura.**)
- Imprima en pantalla su aproximación al número  $\pi$  con 10 dígitos decimales.



**Figure 1:** Scattering para la estimación de  $\pi$  utilizando el método de Monte Carlo.



Se como este Sandía estudioso y pasarás todos tus ramos.