

Tarea 06

9 de octubre de 2019

Primera parte (50 puntos)

Guarde el siguiente código en un archivo llamado `grafica_gauss.py`.

```
1 import matplotlib.pyplot as plt
2 import math
3
4 n_sigma = 10
5 sigma = n_sigma * 0.05
6 n_points = 200
7 deltax = 20/n_points
8
9 l = list(range(n_points))
10 x = []
11 for i in l:
12     x.append(-10.0 + deltax * i)
13
14 y = []
15 for i in x:
16     y.append( math.exp( -0.5*i**2/sigma**2) )
17
18 plt.figure()
19 plt.plot(x,y)
20 plt.savefig( '{}.png'.format(n_sigma) )
21 plt.close()
```

Listing 1: `grafica_gauss.py`

Este código genera `n_points` puntos entre -10 y 10 para graficar una gaussiana centrada en 0 y $\sigma = n_sigma \times 0,05$. La gráfica se guarda como `n_sigma.png`.

- Corra el código una vez para comprobar su funcionamiento.
- Cambie `n_sigma` (entero) para comprobar su funcionamiento.
- Modifique el código para que sea una función de nombre `genera()` que tenga como argumento de entrada `n_sigma`. Con esto, al llamar la `genera(n_sigma)` se genera la gráfica y la guarda.
- Guarde este archivo.

Cree un Notebook llamado `ejercicio06.ipynb`.

- Importe el módulo `grafica_gauss` que creó en el numeral anterior.

- Llame la función que creó en el numeral anterior para comprobar que funciona.
- Inicie un contador n que cambia entre 1 y 50 para llamar a la función `genera(n)`.

Ejecute la celda. Esto debería generar cerca de 49 archivos de imagen.

Ahora vamos a hacer una `pelicula.gif` con las imágenes que se crearon en el numeral anterior.

- En una celda del notebook cree una lista con los nombres de las imágenes como strings. Llámela `filenames`.
- Use una estructura de código similar a la siguiente para que el módulo `imageio` genere un nuevo archivo apilando las imágenes.

```

1 import imageio
2
3 filenames = []  ## Lista con los nombres de los archivos
4
5 with imageio.get_writer("pelicula.gif", mode="I") as writer:
6     for filename in filenames:
7         image = imageio.imread(filename)
8         writer.append_data(image)

```

Listing 2: gif animado

Segunda parte (50 puntos)

En un archivo llamado `grafica_arquimedes.py` escriba el código necesario para tener una función `espiral(n)` que grafica espiral de Arquímedes como la mostrada en Wikipedia https://en.wikipedia.org/wiki/Archimedean_spiral donde el parámetro $b = n/100,0$ y $a = 0$. Las funciones seno y coseno dentro de python las puede utilizar como `math.sin()` y `math.cos()` luego de importar `math`. La gráfica debe quedar guardada en un archivo `png`.

Dentro del notebook `ejercicio06.ipynb`.

- Importe el módulo `grafica_arquimedes` que creó en el numeral anterior.
- Llame la función que creó en el numeral anterior para comprobar que funciona.
- Inicie un contador n que cambia entre 1 y 100 para llamar a la función `espiral(n)`.

Ejecute la celda. Esto debería generar cerca de 100 archivos de imagen. Con estos archivos haga un gif animado `espiral.gif`.