# Lección 3

José David Ruiz Álvarez

josed.ruiz@udea.edu.co

Instituto de Física, Facultad de Ciencias Exactas y Naturales

**Universidad de Antioquia**

28 de agosto de 2018

## 1. Modelando un conjunto de datos

Dado un conjunto de datos, estos pueden ser modelados matemáticamente tratando de ajustar el comportamiento de los datos a una función. El modelo debe:

- Ajustarse lo más posible a los datos (Calidad del ajuste).
- Tener en cuenta la incertidumbre de los datos.

Si bien el modelo es una simple representación matemática de los datos en física damos un paso adicional en el cual este modelo matemático es derivado o esperado de una teoría física en particular dependiendo del fenómeno. En prime lugar vamos a aprender técnicamente cómo hacer un "fit" para luego entrar en los detalles de la incertidumbre asociada a los datos y la calidad del ajuste.

En primer lugar vamos a generar datos aleatoriamente siguiendo una distribución gaussiana:

```
from numpy import sqrt, pi, exp, linspace, save
def gaussian(x, amp, cen, wid):
   return amp * exp(-(x-cen)**2 /wid)


x = linspace(-10,10)
y = gaussian(x, 2.33, 0.21, 1.51)


save("x_coord",x)
save("y_coord",y)


import matplotlib.pyplot as plt
plt.scatter(x, y)
```

```
plt.show()
```

Ahora vamos a abrir los datos que hemos guardado para hacer el ajuste:

```
from numpy import sqrt, pi, exp, linspace, load
def gaussian(x, amp, cen, wid):
   return amp * exp(-(x-cen)**2 /wid)



x=load("x_coord.npy")
y=load("y_coord.npy")

from scipy.optimize import curve_fit
init_vals = [1, 0, 1]     # for [amp, cen, wid]
best_vals, covar = curve_fit(gaussian, x, y, p0=init_vals)
print best_vals

import matplotlib.pyplot as plt
plt.scatter(x, y)
plt.plot(linspace(-10,10,100),gaussian(linspace(-10,10,100),best_vals[0],best_vals[1],best_vals[2]),"r-
plt.show()
```

**Pregunta 1:** ¿Qué tan bien se ajustan los parámetros de la función utilizada a los datos?

**Pregunta 2:** ¿Qué pasa si cambiamos los valores iniciales que le damos al fit?

**Pregunta 3:** ¿Qué pasa si cambiamos la función utilizada para el ajuste por un polinomio de tercer grado?

**Ejercicio 1:** Reemplace la generación de los datos por lo siguiente:

```
from numpy import sqrt, pi, exp, linspace, save, random
def gaussian(x, amp, cen, wid):
   return amp * exp(-(x-cen)**2 /wid)

x = linspace(-10,10)
y = gaussian(x, 2.33, 0.21, 1.51) + random.normal(0, 0.2, len(x))
```

```
save("x_coord_noisy",x)
save("y_coord_noisy",y)

import matplotlib.pyplot as plt
plt.scatter(x, y)
plt.show()
```

y realice de nuevo el ajuste utilizando una función gaussiana. ¿Es igual de bueno el ajuste que en el caso inicial?

**Ejercicio 2:** Descargue los siguientes datos `https://jruizalv.web.cern.ch/jruizalv/CUFICO_Data/Leccion3/x_spectra.npy` `https://jruizalv.web.cern.ch/jruizalv/CUFICO_Data/Leccion3/y_spectra.npy` y trate de encontrar una funcion con la cual modelarlos. Suba sus resultados en forma de "pull request" al repositorio central en la carpeta "Ejercicio_Leccion3".

**Nota:** Si algunos de los paquetes necesarios para esta lección no se encuentra instalado, puede instalarlos siguiendo las instrucciones enÑ `https://www.scipy.org/install.html`.