

# Universidad de Sonora

División de Ciencias Exactas y Naturales

Departamento de Física



Reporte de Actividad 4

## Ajuste de Datos

Estefania Eng Duran

Física Computacional I 2016-1

Hermosillo - 16 de marzo de 2016

## Introducción

En estadística, el análisis de regresión es un proceso para calcular la relación entre variables. Como campo de estudio, incluye muchas técnicas para el modelado y análisis de la relación entre variable dependiente y una o varias variables independientes. El objetivo es estimar una función de las variables independientes llamada la función de regresión. Muchas veces se usa el análisis de regresión para entender cuál de las variables independientes están relacionadas a la variable dependiente y de qué forma.

Entre las muchas técnicas hay métodos estándar como el método de los mínimos cuadrados. Este método busca la solución dentro de una familia de funciones, la que mejor ajuste a los datos. El criterio de mejor ajuste es que la solución minimice la suma de los cuadrados de los residuos. Los residuos son la diferencia entre el valor observado y el valor ajustado proporcionado por el modelo. Se dice que el método de mínimos cuadrados es paramétrico en tanto que la función de regresión se define en términos de un número finito de parámetros desconocidos que son estimados a partir de los datos.

En el lenguaje Python, los paquetes Numpy y Scipy tienen métodos de caja negra para ajustar datos de una dimensión usando mínimos cuadrados en modalidad lineal (Numpy) y no lineal (Scipy). El método `scipy.optimize.leastsq` encuentra el conjunto de parámetros que minimizan la función de error. Este comienza a partir de una primera conjetura y en cada iteración minimiza los datos proporcionados, y así devuelve la lista de los parámetros que mejor se ajusten a los datos

## Actividad

### Ajuste Lineal

Hicimos un ajuste lineal para la colección de datos proporcionada por el sitio web de Quantitative Environmental Learning Project (QELP) de los valores medios de temperatura anuales del estado de Nueva York. Se obtuvieron promediando los valores medios de temperatura anuales de 40 estaciones de clima esparcidos por el estado.

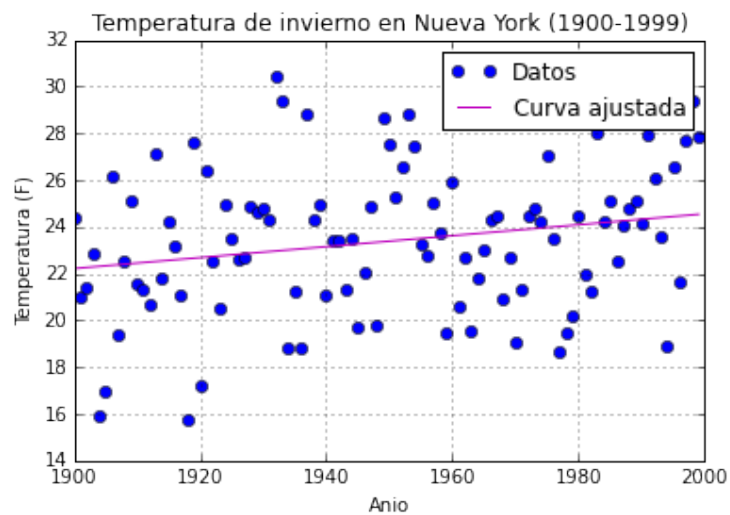
## Código

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import optimize

datos = np.loadtxt('NYWmeantemp.txt')
x=datos[:,0]
y=datos[:,1]

fitfunc = lambda p, x: p[0] + p[1]*x
errfunc = lambda p, x, y: fitfunc(p, x) - y
p0 = [-15., -1.] # Estimación inicial de parametros
p1, success = optimize.leastsq(errfunc, p0[:], args=(x, y))
time = np.linspace(x.min(), x.max(), 100)

plt.plot(x, y, "bo", label="Datos")
plt.plot(time, fitfunc(p1, time), "m-", label="Lineal")
plt.title("Temperatura de invierno en Nueva York (1900-1999)")
plt.grid()
plt.legend()
plt.xlabel("Año")
plt.ylabel("Temperatura (F)")
```



Al correr el código anterior obtuvimos esta gráfica que muestra que los valores de la temperatura media de Nueva York poseen gran variabilidad año con año y se puede decir que la tendencia lineal es débil. Sin embargo, se observa que hay un ligero incremento con respecto al tiempo en las temperaturas.

## **Ajuste Exponencial**

La fuerza gravitacional de la tierra actúa sobre los gases atmosféricos de tal manera que la atmósfera es más densa cerca del nivel de mar y decrece con la altitud. La presión ejercida por la atmósfera incrementa al acercarse uno al suelo debido a que el peso acumulativo de las moléculas de gas que conforman la atmósfera incrementa. Los datos dados por la QUELP no son mediciones de la presión atmosféricas sino valores producidos por la fórmula exponencial de la presión como función de la altitud.

## Código

```
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
import numpy as np

datos = np.loadtxt('earthatm.txt')
x1=datos[:,0]
y1=datos[:,1]

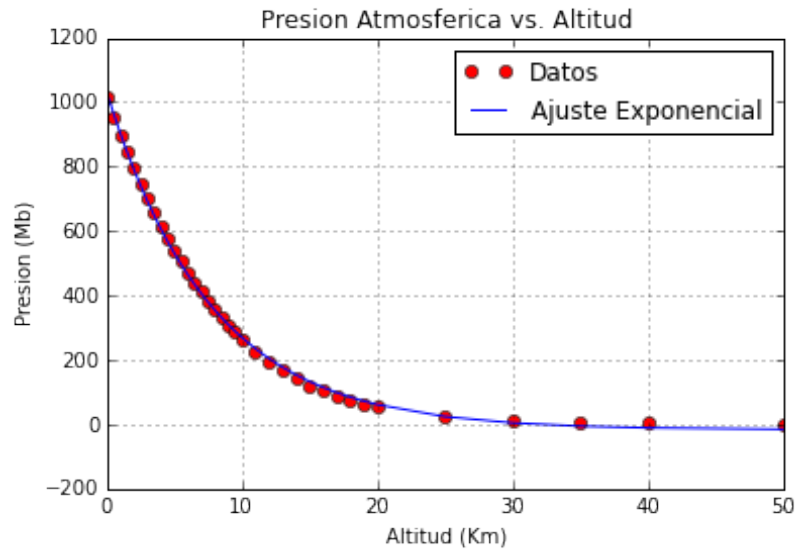
def f(x,a,b,c):
    return c * np.exp(-a * x) + b

popt, pcov = curve_fit(f,x1,y1) #ajuste de curva

plt.plot(x1, y1, "ro", label='Datos')
plt.plot( x1, f(x1, *popt), "b-", label='Exponencial')

plt.grid()
plt.legend()
plt.title("Presion Atmosferica vs. Altitud")
plt.xlabel("Altitud (Km)")
plt.ylabel("Presion (Mb)")

plt.show()
```



Al correr el código anterior obtuvimos esta gráfica. Los datos, al ser producidos por un modelo, no tienen ruido y el ajuste es muy bueno. Se espera que se hagan mediciones reales por parte del QELP para este modelo en corto.

## Referencias

- [1] SciPy CookBook, *Fitting Data*,  
<http://scipy-cookbook.readthedocs.org/items/FittingData.html>