Universidad "Politécnica Salesiana"

"Práctica 2: Simulacion Regresion Lineal"

Alumno: Juan Cañar.
Docente: Ing. Diego Quisi.

Regresión lineal.

En estadística la regresión lineal o ajuste lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente Y, las variables independientes X y un término aleatorio e.

Para hacer una estimación del modelo de regresión lineal simple, trataremos de buscar una recta de la forma: de modo que se ajuste a la nube de puntos. Para esto utilizaremos el método de mínimos cuadrados. Este método consiste en minimizarla suma de los cuadrados de los errores: Es decir, la suma de los cuadrados de las diferencias entre los valores reales observados (yi) y los valores estimados (ˆi y).

Aplicaciones de la regresión lineal.

Una línea de tendencia:

Representa una tendencia en una serie de datos obtenidos a través de un largo período. Este tipo de líneas puede decirnos si un conjunto de datos en particular (como por ejemplo, el PIB, el precio del petróleo o el valor de las acciones) han aumentado o decrementado en un determinado período.

En medicina:

Las primeras evidencias relacionando la mortalidad con el fumar tabaco9 vinieron de estudios que utilizaban la regresión lineal. Los investigadores incluyen una gran cantidad de variables en su análisis de regresión en un esfuerzo por eliminar factores que pudieran producir correlaciones espurias.

En el caso del tabaquismo, los investigadores incluyeron el estado socioeconómico para asegurarse que los efectos de mortalidad por tabaquismo no sean un efecto de su educación o posición económica.

Formula:

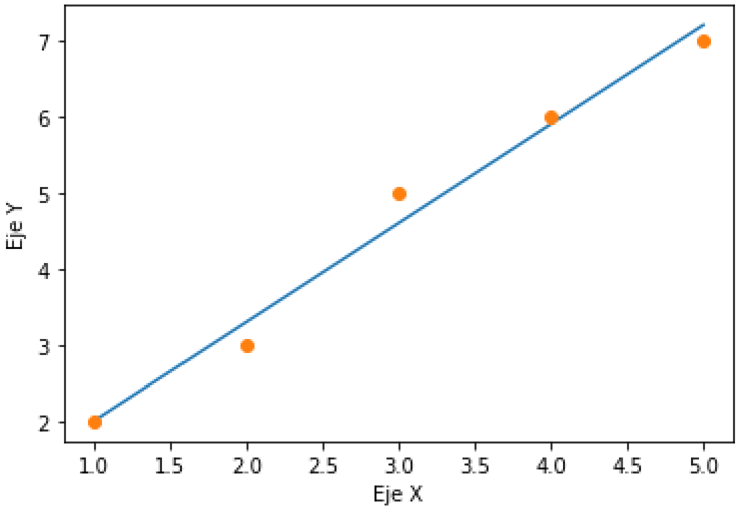
$$\boxed{\hat{\beta}_1 = \frac{cov(x,y)}{s_x^2} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}}$$

$$\boxed{\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}}$$

In [11]:

```
import matplotlib.pyplot as plt
import numpy as np

def promedio(x,y):
    return sum(x) / len(y)

def operacion1(x,y):
    #obtiene x menos el promedio de x
    ter1 = x-np.average(x)
    ter2 = y-np.average(y)
    sxy = sum(ter1*ter2)
    sxx = sum(ter1*ter1)
    resp =sxy/sxx
    #print("--->",resp)
    return resp

def operacion0(x,y):
    res = np.average(y)-operacion1(x,y)*np.average(x)
    #print("->",res)
    return res

def graficar(x,y,z):
    b1 =operacion1(x,y)
    b0 =operacion0(x,y)
    predecir= b1*z+b0
    #print('**.',b1,b0,predecir)
    puntos_x = np.linspace(x[0],x[-1],6)
    puntos_y = b0+b1*puntos_x
    print("Formula aplicada: ",puntos_x)
    print('Z=',b1,'*',z,'+',b0)
    print('__________________________')
    print('Prediccion = *',predecir)
    plt.plot(puntos_x,puntos_y,)
    plt.plot(x,y,"o")
    plt.xlabel('Eje X')
    plt.ylabel('Eje Y')

    plt.show()

if __name__=="__main__":
    print("CALCULO DE LA FUNCION DE REGRESION LINEAL")
    z=cantidad = float(input("ESCRIBA SU EDAD( EJEMPLO 28)): "))
    x=[1,2,3,4,5]
    y=[2,3,5,6,7]
    graficar(x,y,z)
```

```
CALCULO DE LA FUNCION DE REGRESION LINEAL
ESCRIBA SU EDAD( EJEMPLO 28)): 9
Formula aplicada:  [1.  1.8 2.6 3.4 4.2 5. ]
Z= 1.3 * 9.0 + 0.6999999999999993
__________________________
Prediccion = * 12.4
```



Conclusiones:

- Mediante las librerias de numpy y matplotlib se puede representar como es el calculo de la regresion linela , ya que numpy nos facilita ciertos metodos como average para calcular el promedio entre dos vectores y linspace que coje un termino inicial y un termino final para la grafica.
- Mediante el uso de regresion lineal se puede realizar predicciones, dando una variable dependiente Y, y la variables independientes Xi y un término aleatorio e.