Gráfico con Regresión Lineal

LINK: https://github.com/DavidPilataxi/MetodosNumericosGr1cc/blob/main/Taller%2004/Taller04.ipynb

Taller 04

Ajuste de curvas por mínimos cuadrados

David Pilataxi

Gr1cc

14/12/24

```
import numpy as np
from ipywidgets import interact
import matplotlib.pyplot as plt

p1 = (5.4, 3.2)
p3 = (12.3, -3.6)

def calcular_regresion_lineal(x_coords, y_coords):
    # Convertir las listas de coordenadas a arrays de numpy
    x = np.array(x_coords)
    y = np.array(y_coords)

    # Calcular la pendiente (m) y la intersección (b) usando la fórmula de regresión lineal
    m, b = np.polyfit(x, y, 1)
    return m, b

def update_plot(p2_x, p2_y):
    x_coords = [p1[0], p2_x, p3[0]]
```

```
    y_coords = [p1[1], p2_y, p3[1]]

    # Calcular la regresión lineal de los puntos
    m, b = calcular_regresion_lineal(x_coords, y_coords)

    # Graficar los puntos y la recta de regresión
    plt.figure(figsize=(10, 6))
    plt.scatter(x_coords, y_coords, color="red")

    x_line = [min(x_coords), max(x_coords)]
    y_line = [m * x + b for x in x_line]
    plt.plot(x_line, y_line, color="blue")

    plt.xlabel("X")
    plt.ylabel("Y")
    plt.title("Gráfico de puntos y líneas con regresión lineal")
    plt.show()

    # Mostrar los coeficientes de la regresión
    print(f"Regresión lineal: y = {m:.2f}x + {b:.2f}")

_ = interact(update_plot, p2_x=(5.5, 12.3, 0.1), p2_y=(-10.0, 10.0, 0.1))
```

```
interactive(children=(FloatSlider(value=8.9, description='p2_x', max=12.3, min=5.5), FloatSli
```

El renderizado de quarto no procesa grafica con objetos interactivos, por ello se adjuntan distintos ejemplos.

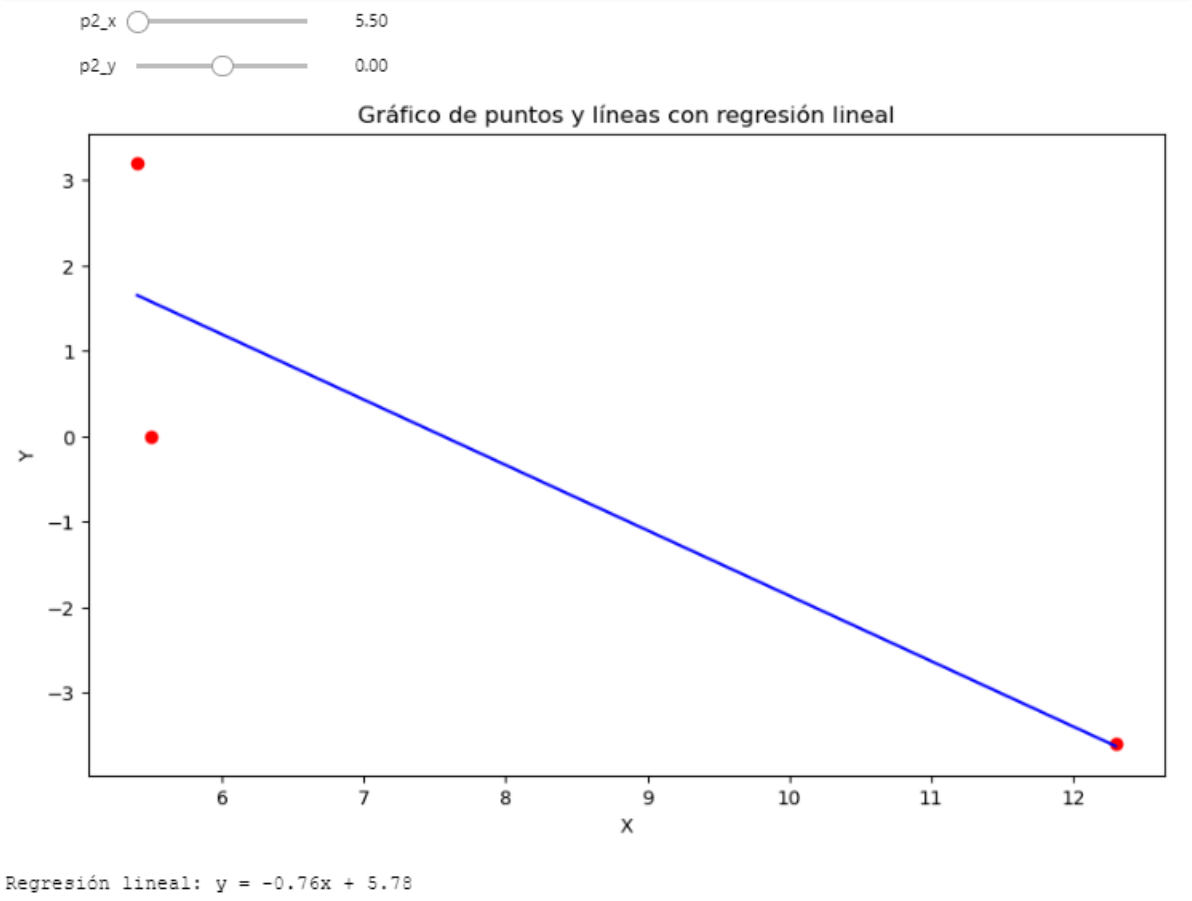

Figure 1: image.png

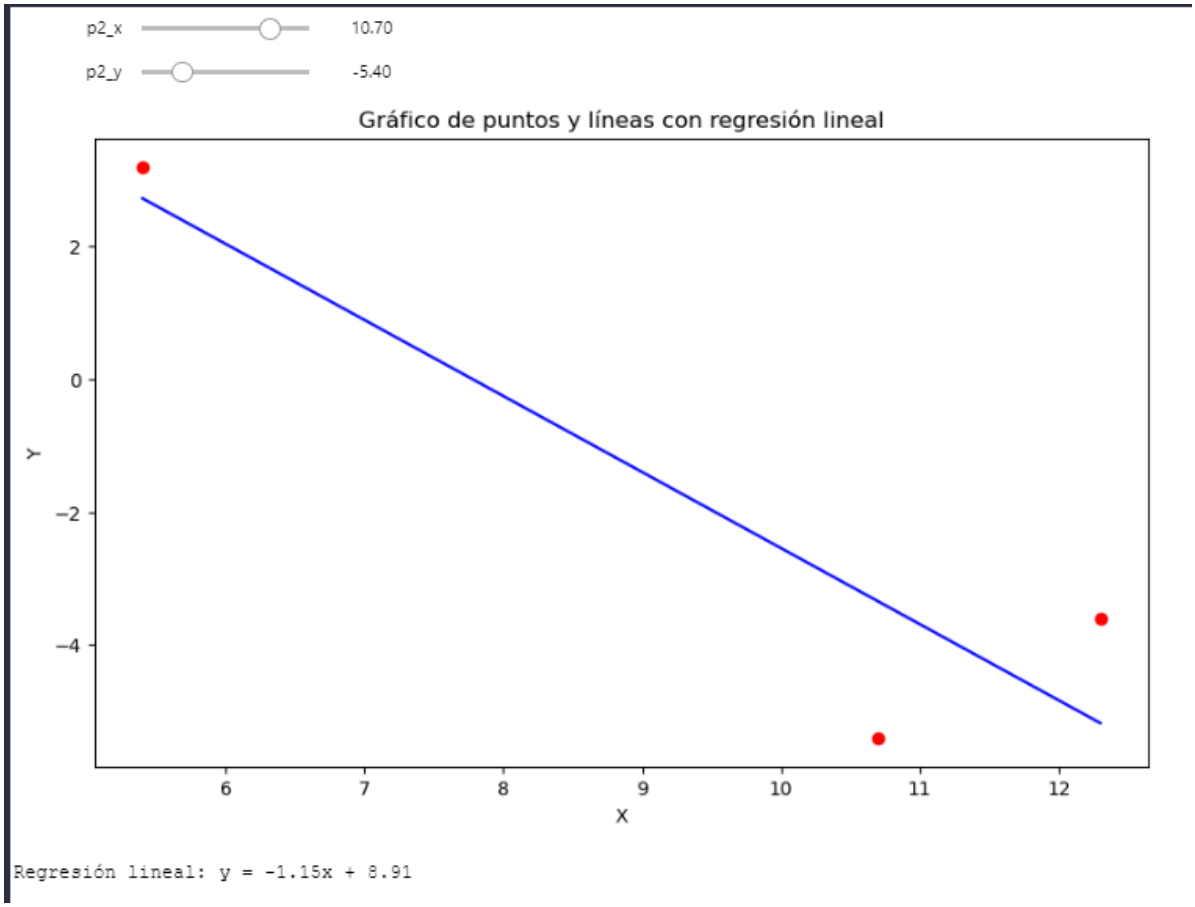

Figure 2: image.png

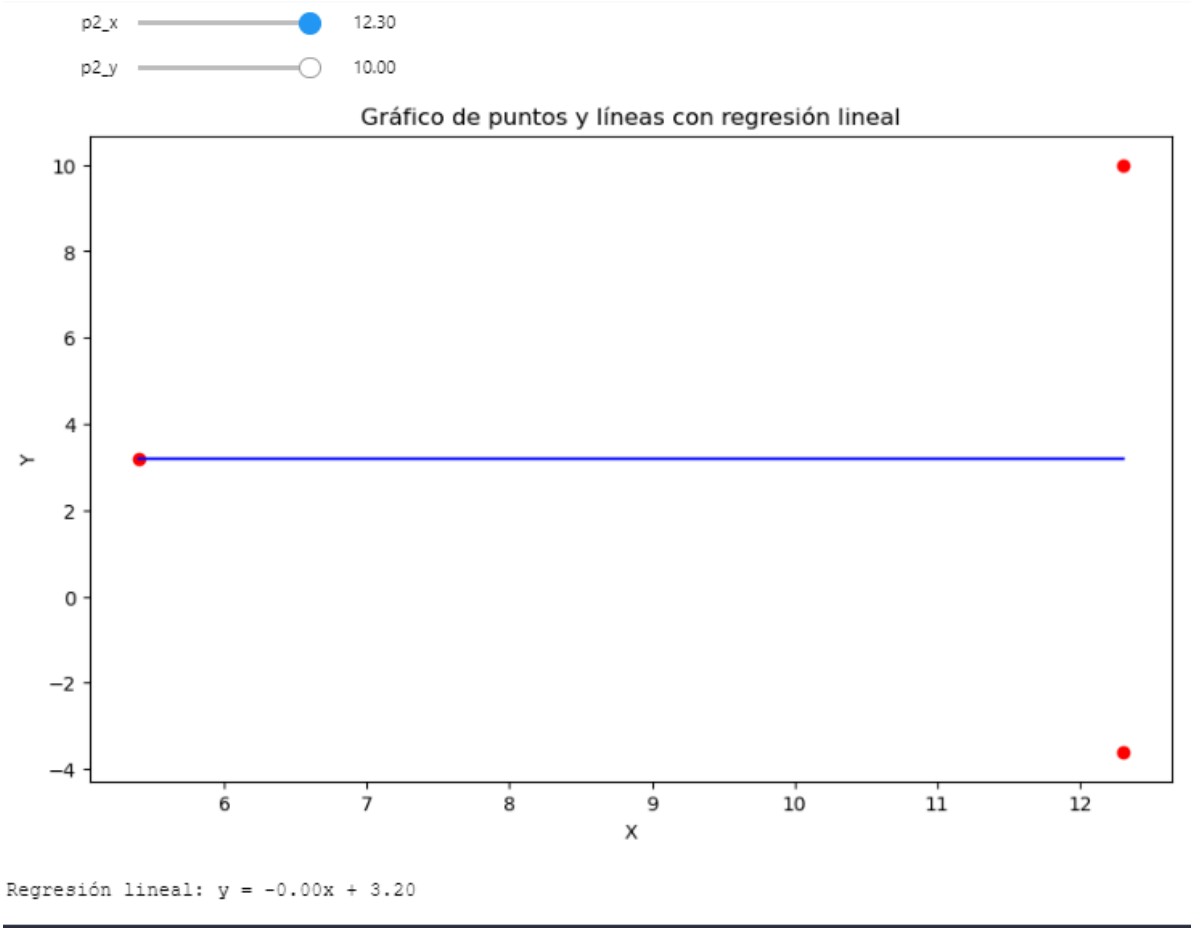

Figure 3: image.png