

Métodos Numéricos -Taller 04- Mínimos Cuadrados

Alicia Pereira

Tabla de Contenidos

1 Taller 04	1
1.1 Ajuste de curvas por mínimos cuadrados	1

1 Taller 04

1.1 Ajuste de curvas por mínimos cuadrados

```
p1 = (-2, 2)
p2_i = (0, -1)
p3 = (2, -2)
p4 =(6,2)
```

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

def update_plot(p2_x, p2_y):
    x_coords = [p1[0], p2_x, p3[0],p4[0]]
    y_coords = [p1[1], p2_y, p3[1],p4[1]]

    m, b = np.polyfit(x_coords, y_coords, 1)

    plt.figure(figsize=(10, 6))
    plt.scatter(x_coords, y_coords, color="orchid", label="Puntos")
```

```

x_line = np.linspace(min(x_coords) - 1, max(x_coords) + 1, 100)
y_line = m * x_line + b
plt.plot(x_line, y_line, color="skyblue", label=f"Línea ajustada:  $y = \{m:.2f\}x + \{b:.2f\}$ ")

plt.xlabel("X")
plt.ylabel("Y")
plt.title("Points and Line Plot")
plt.show()

_ = 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), FloatSl

```

GitHub: [git@github.com: dayapt04](https://github.com/dayapt04)

[GitHub Métodos Numéricos - Repositorio](#)