

La regresión lineal es un método estadístico utilizado para modelar la relación entre una variable independiente (o predictora) y una variable dependiente (o de respuesta) mediante una línea recta. Aquí te proporciono un ejemplo simple de regresión lineal en Python utilizando la biblioteca `scikit-learn`:

```
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Datos de ejemplo (variables independientes x y variables dependientes y)
x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 3, 4, 5, 6])

# Crear un objeto de regresión lineal
modelo = LinearRegression()

# Ajustar el modelo a los datos
modelo.fit(x.reshape(-1, 1), y)

# Realizar predicciones con el modelo entrenado
y_pred = modelo.predict(x.reshape(-1, 1))

# Visualizar los resultados
plt.scatter(x, y, label='Datos reales')
plt.plot(x, y_pred, color='red', label='Regresión lineal')
plt.xlabel('Variable independiente (x)')
plt.ylabel('Variable dependiente (y)')
plt.legend()
plt.show()
```

```
# Imprimir los coeficientes de la línea recta (pendiente e intercepto)
print("Pendiente (coeficiente 'a'): ", modelo.coef_[0])
print("Intercepto (coeficiente 'b'): ", modelo.intercept_)
```

Este código primero crea datos de ejemplo ``x`` e ``y``, luego ajusta un modelo de regresión lineal a estos datos utilizando ``LinearRegression`` de ``scikit-learn``. Luego, realiza predicciones con el modelo entrenado y visualiza los datos originales junto con la línea de regresión resultante. Finalmente, imprime los coeficientes de la línea recta (pendiente e intercepto) que mejor se ajustan a los datos.

Asegúrate de tener ``scikit-learn`` y ``matplotlib`` instalados en tu entorno de Python para ejecutar este código.