

Tarea 1

Estudiante: Sheila Rojas Vasquez

Carrera: Ingeniería de Sistemas

Explicación del código:

El código inicia generando un conjunto de datos aleatorios que simulan las alturas y pesos de 100 personas. Se establece una relación entre ambas variables, donde el peso se calcula de forma aleatoria pero dentro de un rango que depende de la estatura, utilizando el índice de masa corporal (IMC) como referencia desde un peso bajo, peso normal y un peso alto. Estos datos son almacenados en un DataFrame de pandas para facilitar su manipulación. Posteriormente, se calcula la ecuación de una recta que mejor se ajusta a estos datos, utilizando el método de mínimos cuadrados. Esta recta representa la relación lineal entre la estatura y el peso.

Una vez calculada la ecuación de la recta, se generan los valores de y (peso) correspondientes a cada valor de x (estatura) según la ecuación obtenida. Finalmente, se crea un gráfico de dispersión donde se representan los datos originales (altura vs peso) y la recta ajustada. Este gráfico permite visualizar la relación entre ambas variables y evaluar la calidad del ajuste del modelo lineal. La recta ajustada puede utilizarse para realizar predicciones de peso a partir de una estatura dada, dentro del rango de valores utilizados para entrenar el modelo.

Código:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Semilla para la reproducibilidad de los resultados
np.random.seed(42)
alturas = np.random.uniform(1.4, 2.0, 100) # Generamos 100 estaturas
aleatorias entre 1.4m y 2.0m
pesos = []
# Bucle para generar pesos aleatorios controlados según la estatura
for altura in alturas:
    # Calculo del peso mínimo y máximo usando el IMC
    peso_min = 18.5 * (altura**2)
    peso_max = 29.9 * (altura**2)
    # Generar un peso aleatorio entre el peso mínimo y máximo
    peso = np.random.uniform(peso_min, peso_max)
    pesos.append(peso)
data = pd.DataFrame({
    'Estatura (m)': alturas,
    'Peso (kg)': pesos
})
```

```

# Calcular la pendiente (m) y la intersección (b) de la recta  $y = mx + b$ 
x = data['Estatura (m)']
y = data['Peso (kg)']
m = np.sum((x - np.mean(x)) * (y - np.mean(y))) / np.sum((x - np.mean(x))
** 2)
b = np.mean(y) - m * np.mean(x)

# Crear los valores de y basados en la fórmula de la recta
y_linea = m * x + b

# Gráfico de los datos generados y la recta ajustada
plt.scatter(data['Estatura (m)'], data['Peso (kg)'], color='green',
Label='Datos')
plt.plot(x, y_linea, color='blue', Label='Línea ajustada')
plt.title('Estatura vs Peso con Línea Ajustada')
plt.xlabel('Estatura (m)')
plt.ylabel('Peso (kg)')
plt.legend()
plt.show()

```

