# -\*- coding: utf-8 -\*-

"""

Created on Mon Apr 29 11:05:51 2024

@author: ELISHA JOY R. YUMANG

"""

import numpy as np

import matplotlib.pyplot as plt

# Generate some sample data

np.random.seed(42)

X = np.linspace(0, 10, 100)

true\_slope = 2

true\_intercept = 5

y = true\_slope \* X + true\_intercept + np.random.normal(0, 2, 100)

# Define the Bayesian linear regression model

def bayesian\_linear\_regression(X, y, num\_samples=1000):

# Define prior hyperparameters

prior\_mean = np.zeros(2)

prior\_cov = np.eye(2) \* 0.1

# Compute posterior parameters

X\_b = np.c\_[np.ones\_like(X), X]

posterior\_cov = np.linalg.inv(prior\_cov + X\_b.T @ X\_b)

posterior\_mean = posterior\_cov @ (prior\_cov @ prior\_mean + X\_b.T @ y)

# Sample from the posterior distribution

samples = np.random.multivariate\_normal(posterior\_mean, posterior\_cov, size=num\_samples)

return samples

# Perform Bayesian linear regression

samples = bayesian\_linear\_regression(X, y)

# Plot the results

plt.figure(figsize=(10, 6))

plt.scatter(X, y, label='Data')

for i in range(50):

y\_pred = samples[i, 0] + samples[i, 1] \* X

plt.plot(X, y\_pred, color='gray', alpha=0.1)

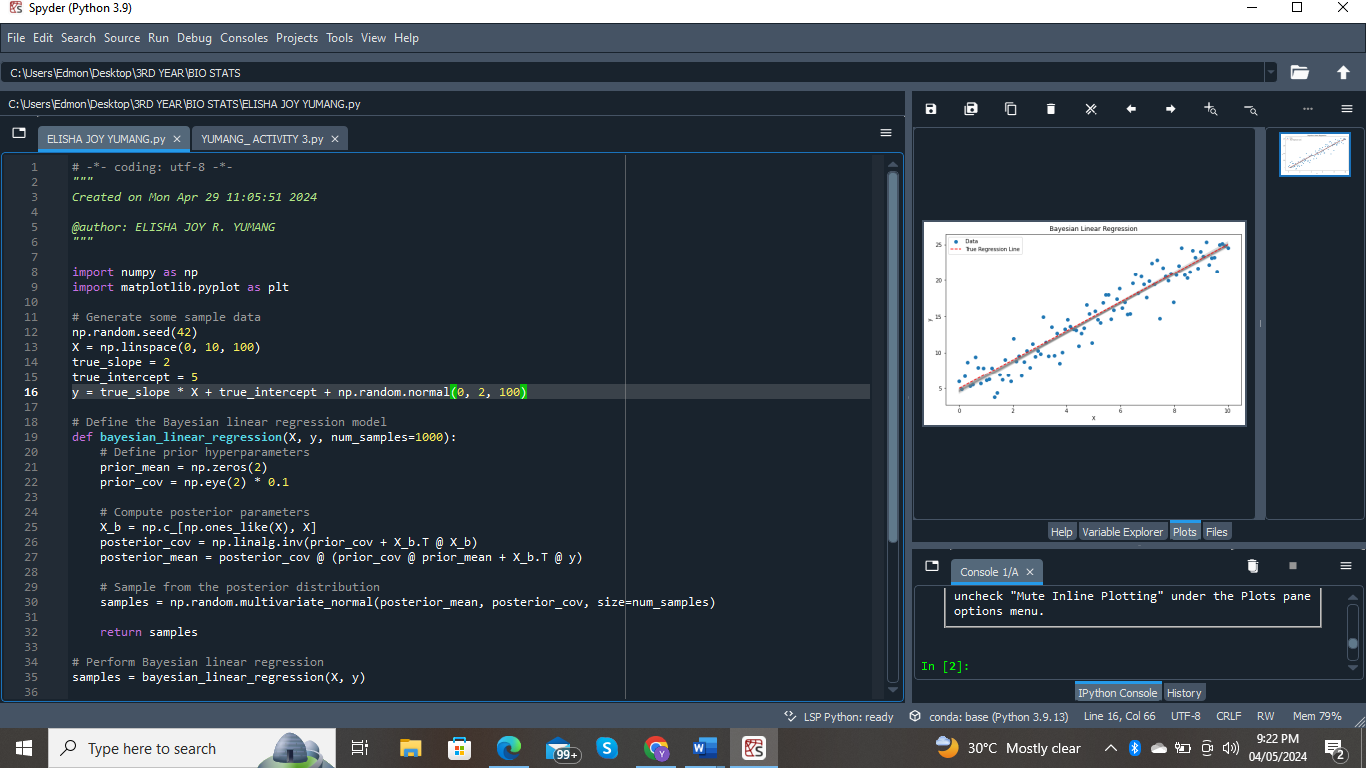
plt.plot(X, true\_slope \* X + true\_intercept, color='r', linestyle='--', label='True Regression Line')

plt.xlabel('X')

plt.ylabel('y')

plt.title('Bayesian Linear Regression')

plt.legend()

plt.show()

