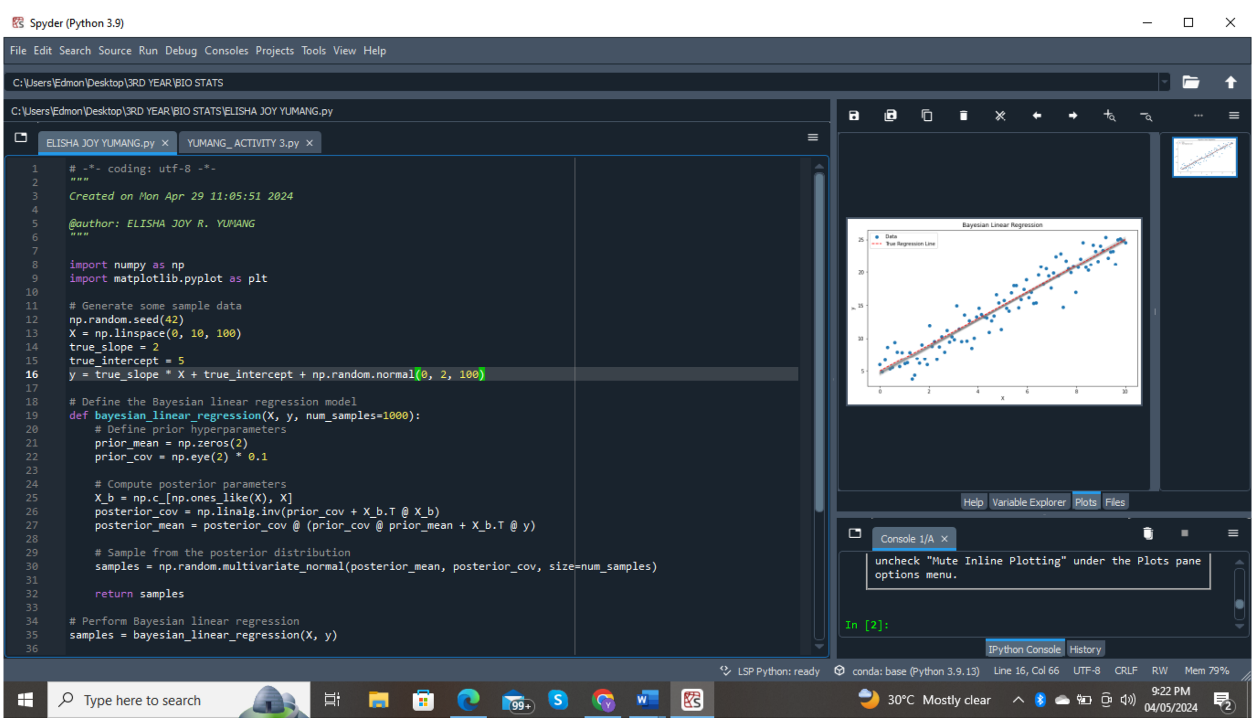
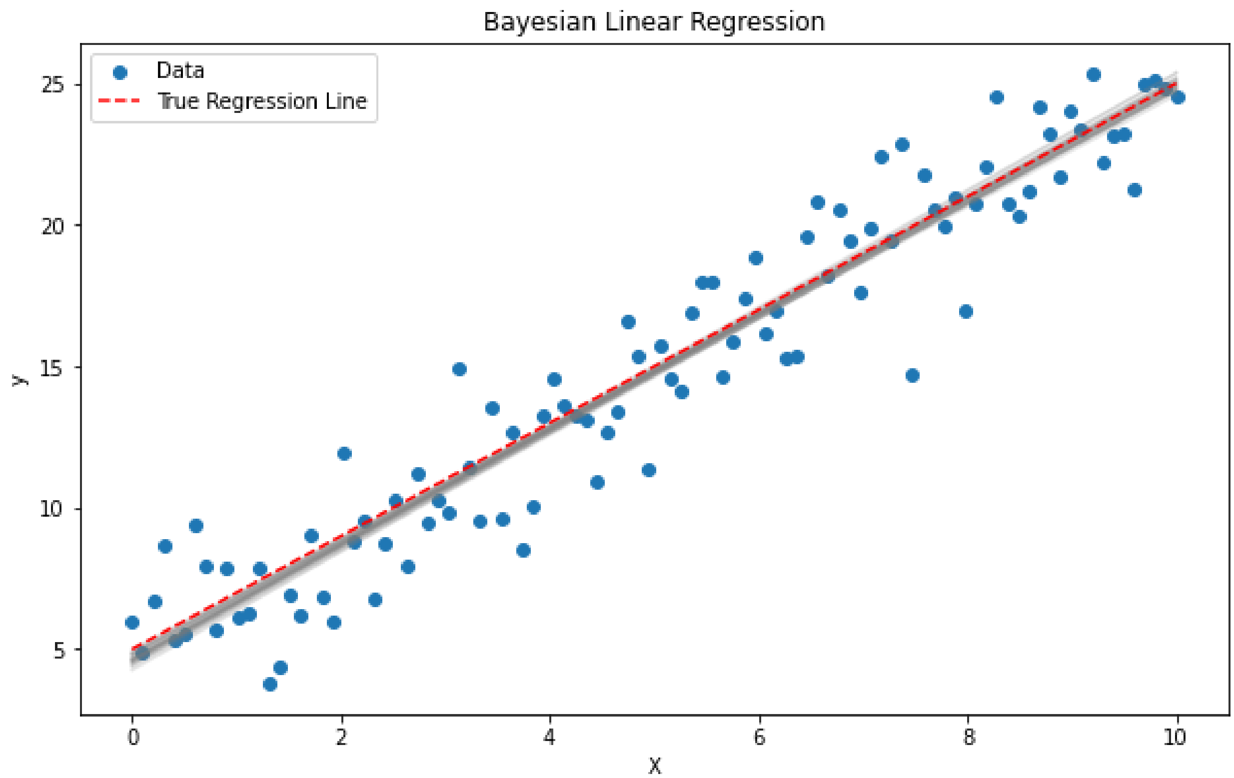
```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 29 11:05:51 2024

@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()
```




Bayesian Linear Regression
Data
True Regression Line
25
20
15
10
5
y
0
2
4
6
8
10
X