

## Experiment - 3A Linear Regression: BMI vs Life Expectancy

### Code

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Load the dataset (replace with the correct path to your file)
path = "Life Expectancy Data.csv"
data = pd.read_csv(path)

# Selecting relevant columns: 'BMI' (independent variable) and 'Life expectancy' (dependent variable)
data = data[['BMI', 'Life expectancy']]
data = data.dropna() # Drop rows with missing values

# Features (X) and target (y)
X = data['BMI'].values.reshape(-1, 1) # Reshape for sklearn compatibility
y = data['Life expectancy'].values

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creating and training the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Making predictions on the test set
y_pred = model.predict(X_test)

# Calculating root mean squared error
rmse = mean_squared_error(y_test, y_pred)**0.5
print(f"Root Mean Squared Error: {rmse:.2f} years")

# Plotting the results
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='blue', label='Data Points') # Scatter plot of data points
plt.plot(X_test, model.predict(X_test), color='red', label='Regression Line')
# Regression line
plt.title('Linear Regression: BMI vs Life Expectancy')
plt.xlabel('BMI')
plt.ylabel('Life Expectancy')
plt.legend()
```

```
plt.savefig("linear_regression_plot.png") # Save the plot as an image file  
plt.show()
```

## Output

