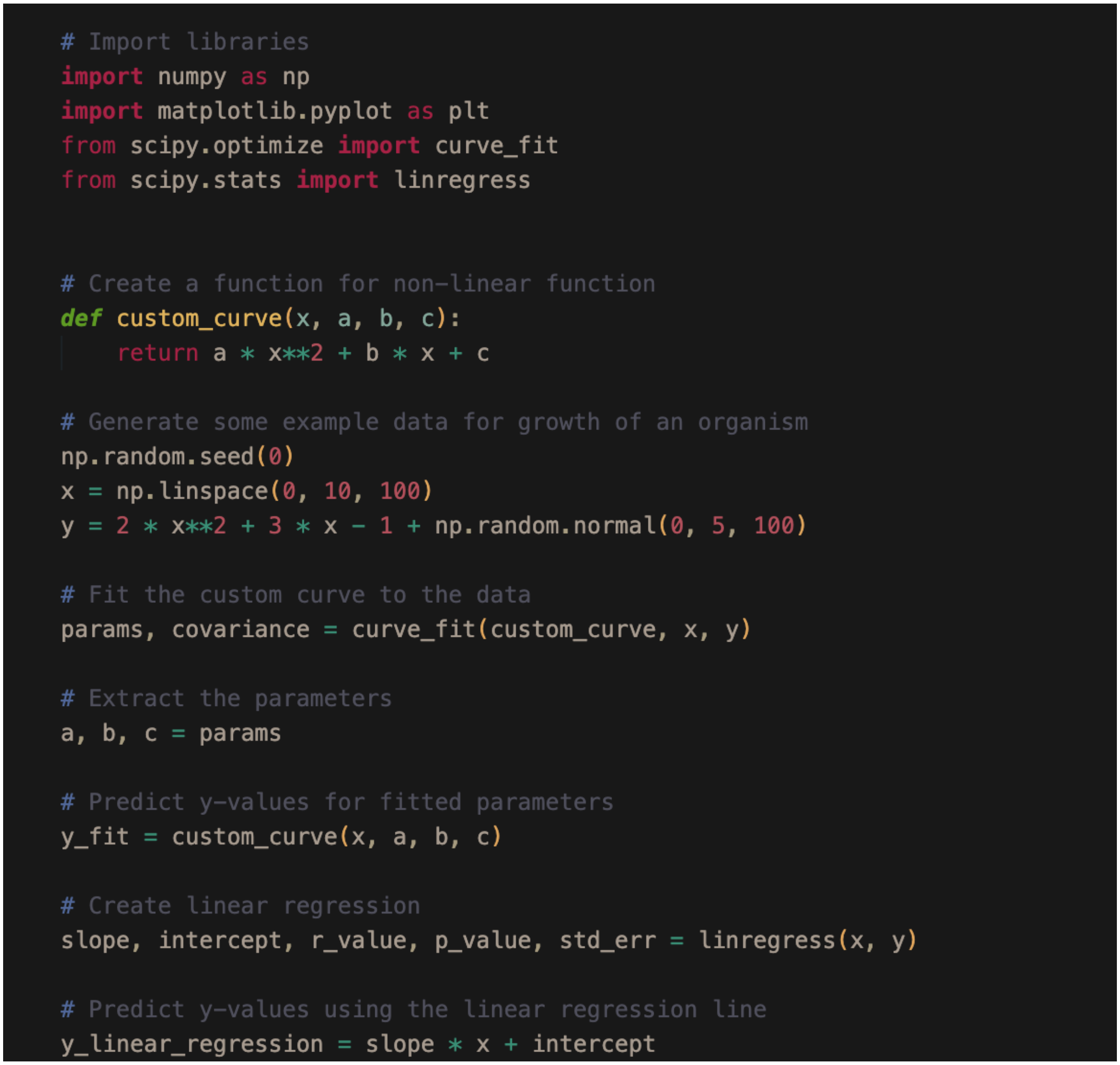
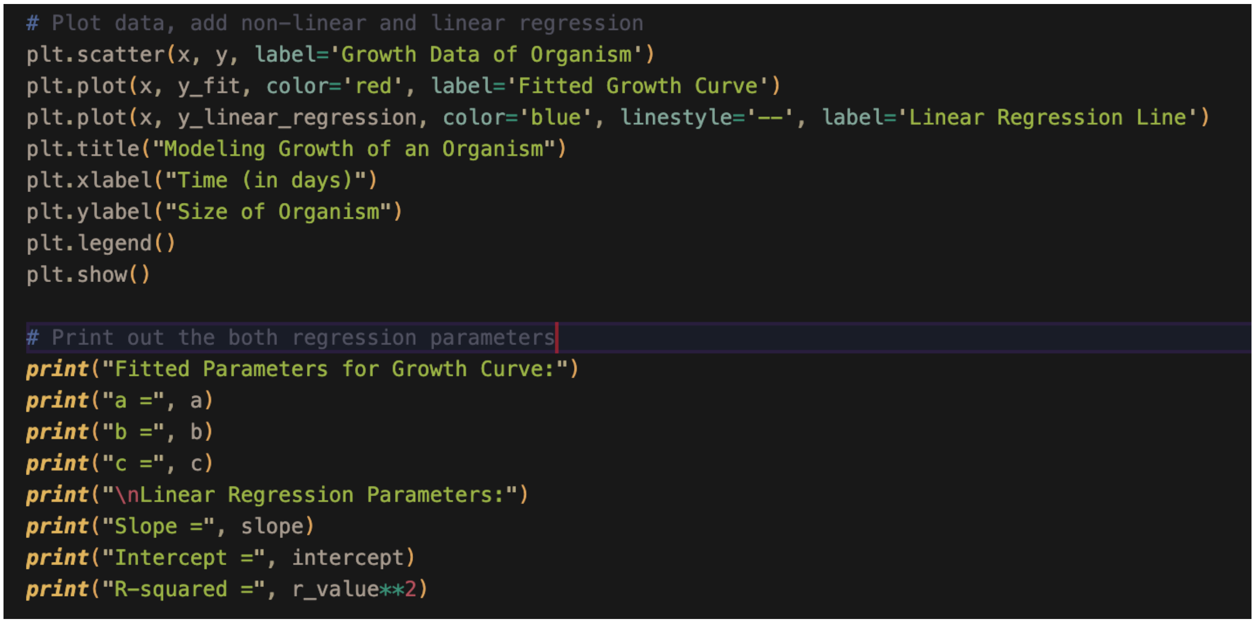
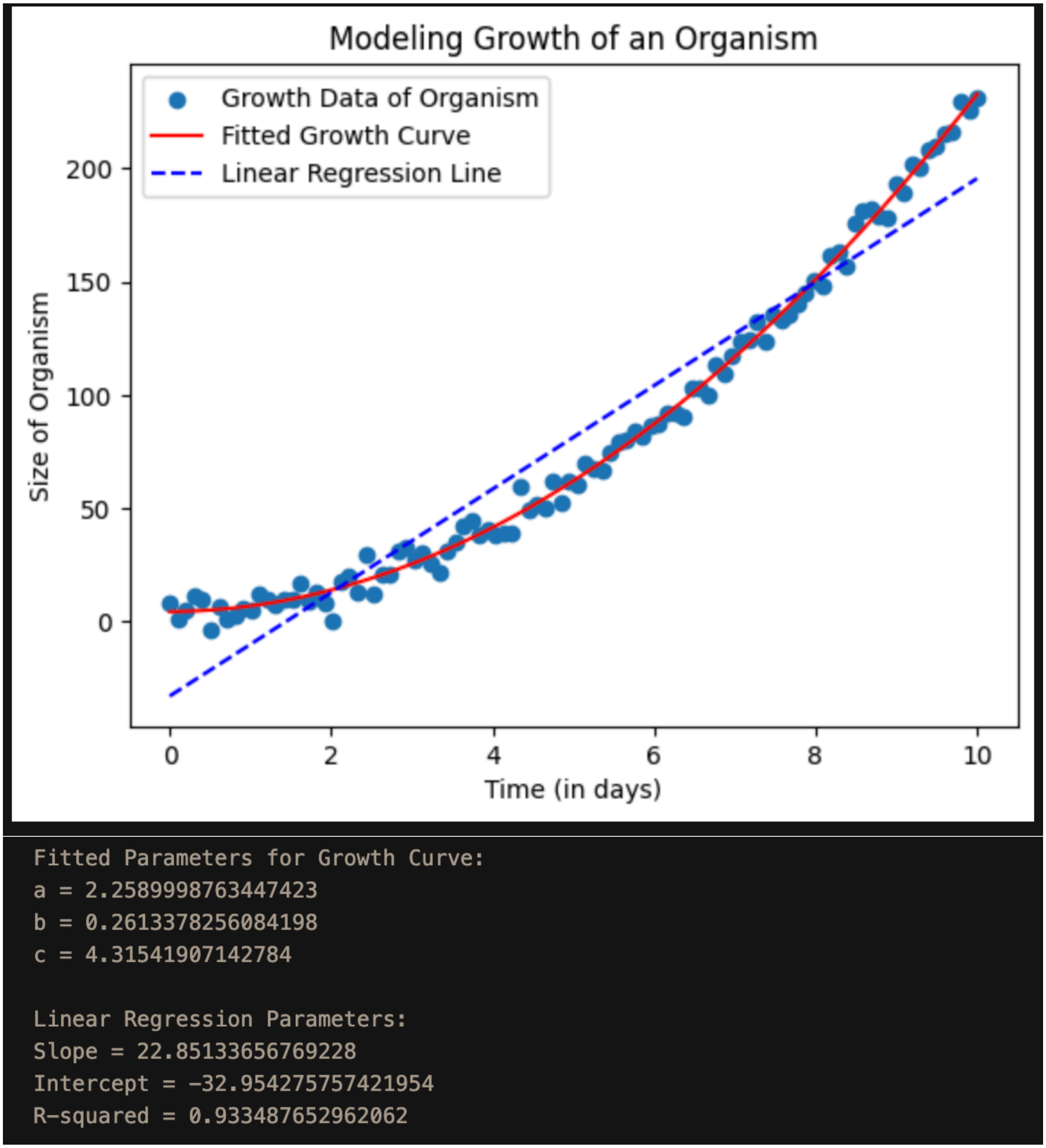
T5D2 Discussion



```
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.stats import linregress


# Create a function for non-linear function
def custom_curve(x, a, b, c):
    return a * x**2 + b * x + c

# Generate some example data for growth of an organism
np.random.seed(0)
x = np.linspace(0, 10, 100)
y = 2 * x**2 + 3 * x - 1 + np.random.normal(0, 5, 100)

# Fit the custom curve to the data
params, covariance = curve_fit(custom_curve, x, y)

# Extract the parameters
a, b, c = params

# Predict y-values for fitted parameters
y_fit = custom_curve(x, a, b, c)

# Create linear regression
slope, intercept, r_value, p_value, std_err = linregress(x, y)

# Predict y-values using the linear regression line
y_linear_regression = slope * x + intercept
```



```
# Plot data, add non-linear and linear regression
plt.scatter(x, y, label='Growth Data of Organism')
plt.plot(x, y_fit, color='red', label='Fitted Growth Curve')
plt.plot(x, y_linear_regression, color='blue', linestyle='--', label='Linear Regression Line')
plt.title("Modeling Growth of an Organism")
plt.xlabel("Time (in days)")
plt.ylabel("Size of Organism")
plt.legend()
plt.show()

# Print out the both regression parameters
print("Fitted Parameters for Growth Curve:")
print("a =", a)
print("b =", b)
print("c =", c)
print("\nLinear Regression Parameters:")
print("Slope =", slope)
print("Intercept =", intercept)
print("R-squared =", r_value**2)
```



```
Fitted Parameters for Growth Curve:
a = 2.2589998763447423
b = 0.2613378256084198
c = 4.31541907142784

Linear Regression Parameters:
Slope = 22.85133656769228
Intercept = -32.954275757421954
R-squared = 0.933487652962062
```