# BIO 171 - Spectrophotometry Lab

June 13, 2024

## 1 Spectrophotometry Lab Data

This notebook contains data collected from a series of spectrophotometry experiments. The data includes measurements of absorbance at various concentrations of a chemical solution. The goal is to analyze the relationship between concentration and absorbance, and to determine the linear regression equation and the ( R^2 ) value.

**Data Description:**

- **Concentration (Molarity):** The concentration of the chemical solution in moles per liter (M).
- **Absorbance:** The absorbance measured using a spectrophotometer, which indicates how much light is absorbed by the solution.

The data points are as follows:

- Concentration: [0.00125, 0.0125, 0.125, 0.25, 0.50, 1.00] M
- Absorbance: [0.001, 0.02, 0.026, 0.061, 0.095, 0.218]

[10]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.linear_model import LinearRegression
```

[2]:
```
data = {
    'Concentration (M)': [1.00, 0.50, 0.25, 0.125, 0.0125, 0.00125],
    'Absorbance': [0.218, 0.095, 0.061, 0.026, 0.020, 0.001],
    'Transmittence (%)': [60.5, 80.4, 86.9, 94.2, 99.4, 99.8]
}

df = pd.DataFrame(data)

# Set the index to start at 1
df.index = pd.RangeIndex(start=1, stop=len(df)+1, step=1)

display(df)
```

| | Concentration (M) | Absorbance | Transmittence (%) |
|---|---|---|---|
| 1 | 1.00000 | 0.218 | 60.5 |
| 2 | 0.50000 | 0.095 | 80.4 |
| 3 | 0.25000 | 0.061 | 86.9 |

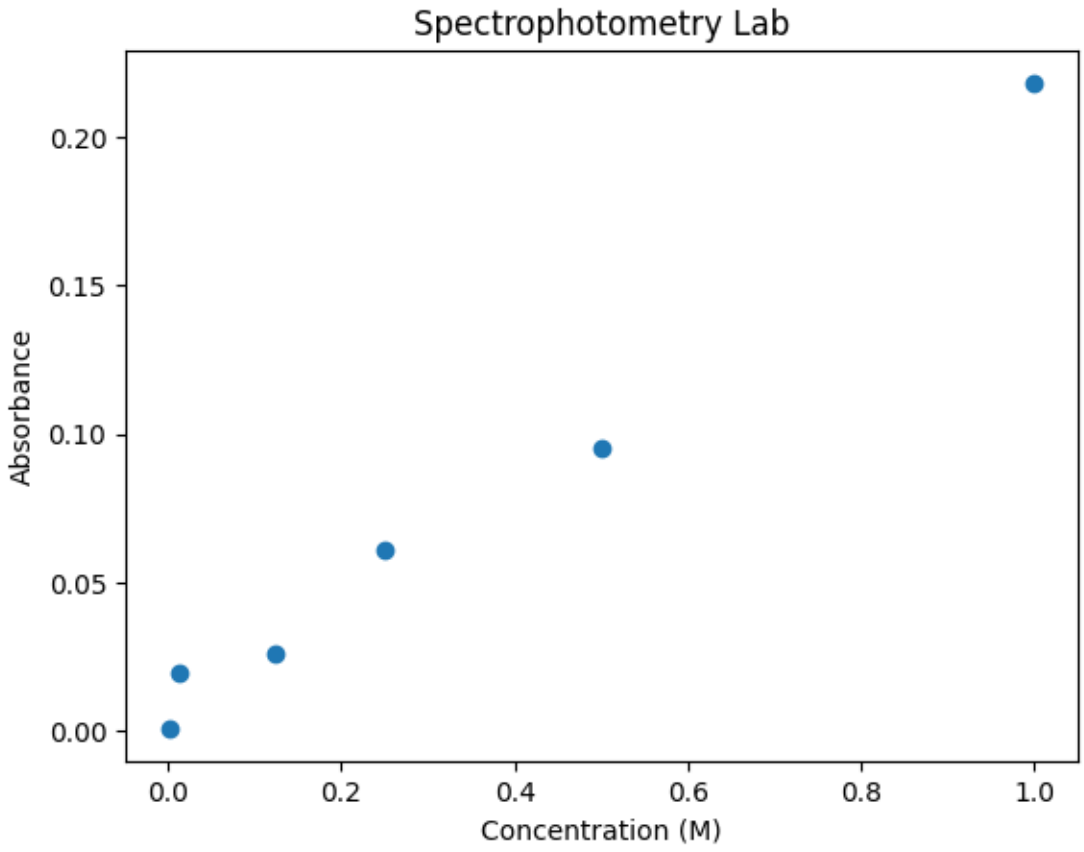
| | | | |
|---|---|---|---|
| 4 | 0.12500 | 0.026 | 94.2 |
| 5 | 0.01250 | 0.020 | 99.4 |
| 6 | 0.00125 | 0.001 | 99.8 |

[3]:
```
# Data
s = [6, 5, 4, 3, 2, 1] # Sample Number
m = [0.00125, 0.0125, 0.125, 0.25, 0.50, 1.00] # Molarity (Concentration)
a = [0.001, 0.02, 0.026, 0.061, 0.095, 0.218] # Absorbance
t = [99.8, 99.4, 94.2, 86.9, 80.4, 60.5] # Transmittance (%)

# Unknown A: a = 0.102, t = 79.3
# Unknown B: a =  0.05, t = 89.2

plt.scatter(m, a)
plt.title("Spectrophotometry Lab")
plt.xlabel("Concentration (M)")
plt.ylabel("Absorbance")

plt.show()
```



[9]:
```
x = [0.00125, 0.0125, 0.125, 0.25, 0.50, 1.00] # Molarity (Concentration)
y = [0.001, 0.02, 0.026, 0.061, 0.095, 0.218] # Absorbance

# Convert x to a NumPy array and reshape it
x = np.array(x).reshape(-1, 1)

# Create a linear regression model
model = LinearRegression()

# Fit the model to the data
model.fit(x, y)

# Predict values
y_pred = model.predict(x)

# Create scatter plot
plt.scatter(x, y, color='blue', label='Data Points')

# Plot the linear regression line
plt.plot(x, y_pred, color='red', label='Linear Regression Line')

# Add labels and title
plt.title("Spectrophotometry Lab")
plt.xlabel("Concentration (M)")
plt.ylabel("Absorbance")
plt.legend()

# Print the equation of the line
slope = model.coef_[0]
intercept = model.intercept_
print(f"The equation of the linear regression line is: y = {slope:.3f}x +␣
 ↪{intercept:.5f}")

# Calculate and print the R^2 value
r_squared = model.score(x, y)
print(f"The R^2 value is: {r_squared:.3f}")

# Add text to the plot with reduced space
equation_text = f"y = {slope:.3f}x + {intercept:.5f}"
r_squared_text = f"R^2 = {r_squared:.3f}"

# Get the axis limits
x_limits = plt.gca().get_xlim()
y_limits = plt.gca().get_ylim()

# Calculate positions for the text based on the axis limits
x_text_position = x_limits[0] + (x_limits[1] - x_limits[0]) * 0.035
```

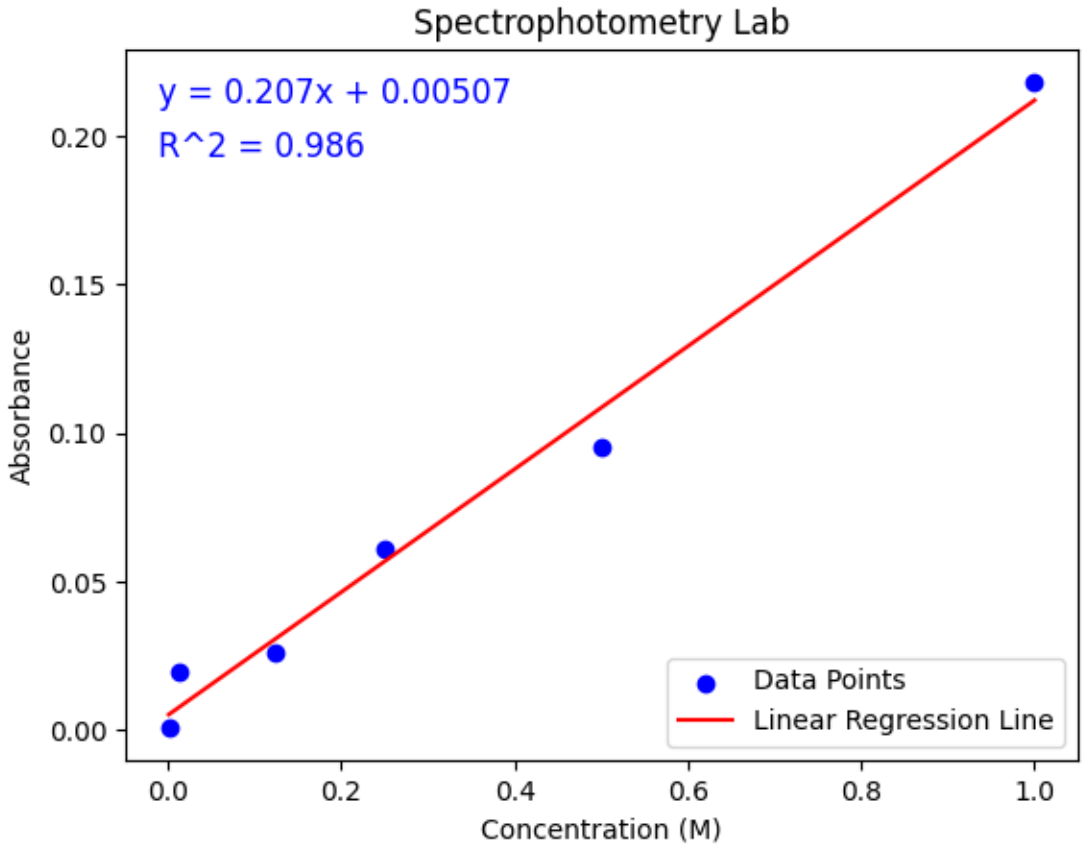
```
y_text_position = y_limits[1] - (y_limits[1] - y_limits[0]) * 0.075

plt.text(x_text_position, y_text_position, equation_text, fontsize=12,␣
 ↪color='blue')
plt.text(x_text_position, y_text_position - (y_limits[1] - y_limits[0]) * 0.
 ↪075, r_squared_text, fontsize=12, color='blue')

# Show the plot
plt.show()

# Print the equation of the line
intercept = model.intercept_
print(f"The equation of the linear regression line is: y = {slope:.3f}x +␣
 ↪{intercept:.5f}")
```

```
The equation of the linear regression line is: y = 0.207x + 0.00507
The R^2 value is: 0.986
```




```
The equation of the linear regression line is: y = 0.207x + 0.00507
```

[ ]: