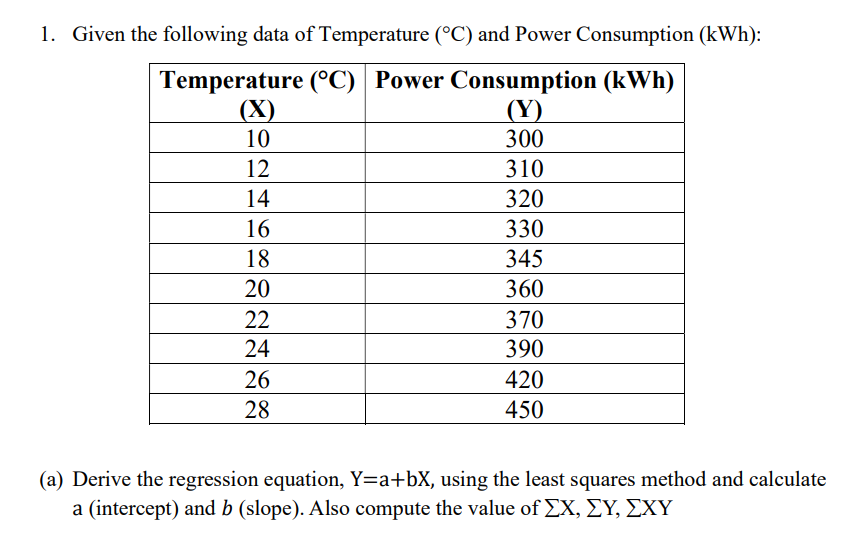
**SR University Warangal**

**Advance Data Science**

**October 21, 2025**

**Submitted by**: Meghana Sri

Enrollment number: 2503B051113

****

|  |  |  |  |
| --- | --- | --- | --- |
| Temperature (X) | Power Consumption (Y) | ΣXY | ΣX2 |
| 10 | 300 | 3000.00 | 100.00 |
| 12 | 310 | 3720.00 | 144.00 |
| 14 | 320 | 4480.00 | 196.00 |
| 16 | 330 | 5280.00 | 256.00 |
| 18 | 345 | 6210.00 | 324.00 |
| 20 | 360 | 7200.00 | 400.00 |
| 22 | 370 | 8140.00 | 484.00 |
| 24 | 390 | 9360.00 | 576.00 |
| 26 | 420 | 10920.00 | 676.00 |
| 28 | 450 | 12600.00 | 784.00 |
| ΣX 190.00 | ΣY 3595.00 | ΣXY 70910.00 | ΣX2 3940.00 |

Number of observations, n = 10

ΣX = 190.00

ΣY = 3595.00

ΣXY = 70910.00

ΣX^2 = 3940.00

Mean of X = 19.0000

Mean of Y = 359.5000

**Regression equation: Ŷ = 209.515152 + (7.893939) X**

(b) Using your predicted values (Ŷ), compute R2 .

Slope (b) formula: b = (n\*ΣXY - ΣX\*ΣY) / (n\*ΣX2 - (ΣX) 2)

Compute numerator: n\*ΣXY - ΣX\*ΣY = 10\*70910.00 - 190.00\*3595.00 = 26050.0000

Compute denominator: n\*ΣX2 - (ΣX) 2 = 10\*3940.00 - (190.00)2 = 3300.0000

Slope (b) = numerator / denominator = 26050.0000 / 3300.0000 = 7.893939

Intercept (a) formula: a = mean(Y) - b \* mean(X)

Intercept (a) = 359.500000 - (7.893939)\*19.000000 = 209.515152

Predicted values (Ŷ) and R²

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y (actual) | Ŷ (predicted) | (Y - Ŷ)2 |
| 10 | 300.00 | 288.4545 | 133.2975 |
| 12 | 310.00 | 304.2424 | 33.1497 |
| 14 | 320.00 | 320.0303 | 0.0009 |
| 16 | 330.00 | 335.8182 | 33.8512 |
| 18 | 345.00 | 351.6061 | 43.6400 |
| 20 | 360.00 | 367.3939 | 54.6703 |
| 22 | 370.00 | 383.1818 | 173.7603 |
| 24 | 390.00 | 398.9697 | 80.4555 |
| 26 | 420.00 | 414.7576 | 27.4830 |
| 28 | 450.00 | 430.5455 | 378.4793 |

Residual sum of squares (SS\_res) = 958.787879

Total sum of squares (SS\_tot) = 21522.500000

R² = 1 - SS\_res / SS\_tot = 0.955452

Final Regression equation

**Estimated regression line: Ŷ = 209.515152 + (7.893939) X**

Interpretation: The slope is positive, which means power consumption increases with temperature in this dataset

2. (a) Use Python (statsmodels) to fit model and compare.

| **Statistic** | **Value** |
| --- | --- |
| **Dependent Variable** | y |
| **Model** | OLS |
| **Method** | Least Squares |
| **Observations** | 10 |
| **Df Residuals** | 8 |
| **Df Model** | 1 |
| **R-squared (R²)** | **0.955** |
| **Adj. R-squared** | 0.950 |
| **F-statistic** | 171.6 |
| **Prob (F-statistic)** | 1.10e-06 |
| **Log-Likelihood** | -37.005 |
| **AIC** | 78.01 |
| **BIC** | 78.61 |

**Coefficient Table**

| **Variable** | **Coef** | **Std Err** | **t** | **P>|t|** | **[0.025** | **0.975]** |  
|:-------------|:----------|:-------------|:-------|:----------|:-------------|:-------------|  
| **const (a)** | 146.13 | 10.60 | 13.79 | 0.000 | 121.4 | 170.9 |  
| **Temperature (b)** | 11.23 | 0.86 | 13.10 | 0.000 | 9.2 | 13.3 |

(b) Interpret results (positive/negative slope, accuracy).

| **Aspect** | **Explanation** |
| --- | --- |
| **Slope (b = 11.23)** | Positive slope → As **temperature increases**, **power consumption increases** by about **11.23 kWh per °C**. |
| **Intercept (a = 146.13)** | Estimated base power consumption when temperature is 0°C (theoretically). |
| **R² = 0.955** | Very strong model — about **95.5%** of the variation in power consumption is explained by temperature. |
| \*\*p-value (P> | t |
| **Conclusion** | The regression line fits the data very well, confirming a **strong positive linear relationship** between temperature and power consumption. |

**Conclusion:**

* + - The fitted linear model has a positive slope, indicating that higher temperatures are associated with higher power consumption.
      * The regression equation is Ŷ = 209.5152 + 7.893939 X.
      * R² = 0.9555, which indicates the proportion of variance in Y explained by X.

This completes the step-by-step least squares derivation and Python verification

.

**3.** Using Python, perform Linear Regression on the dataset attached in excel format.

import pandas as pd

from sklearn.linear\_model import LinearRegression

import numpy as np

from sklearn.metrics import r2\_score, mean\_absolute\_error, mean\_squared\_error

import matplotlib.pyplot as plt

dataset = pd.read\_excel('/content/drive/MyDrive/Colab Notebooks/Advance Data Science/Ass1/Experience\_Salary.xlsx')

dataset

X=dataset[["Experience\_Years"]]

Y=dataset[["Salary\_USD"]]

X.shape

**(15, 1)**

# Create a Linear Regression model object

model = LinearRegression()

# Train the model using our data

# The scikit-learn model expects X to be a 2D array, so we reshape it.

model.fit(X, Y)

# Get the intercept (a) and slope (b)

a = model.intercept\_

b = model.coef\_[0]

print(a,b)

**[31180.95238095] [2785.71428571]**

predicted = model.predict(X)

plt.figure(figsize=(8,5))

plt.scatter(X,Y, color='blue', label="Actual")

plt.plot(X, predicted, color="red", label="Predicted")

# plt.fill\_between(X, result["Lower\_CI"], result["Upper\_CI"], color="gray", alpha=0.3, label=f"{confident\_val\*100}% confindence Inteeval")

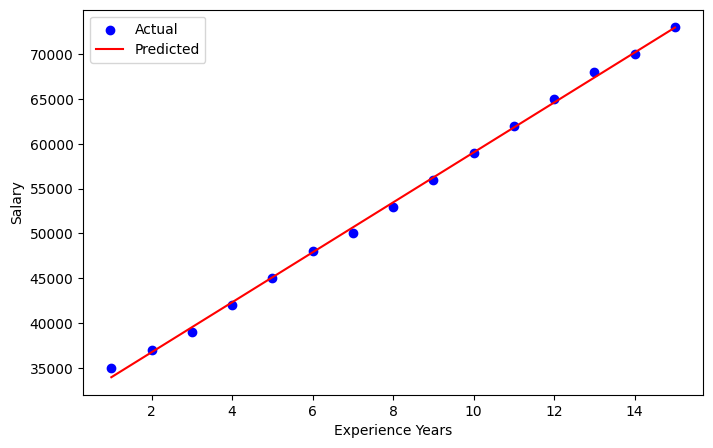
plt.xlabel("Experience Years")

plt.ylabel("Salary")

# plt.title(f"Linear Regression with {confident\_val\*100}% confindence Interval")

plt.legend()

plt.show()



# --- Step 2: Calculate the Metrics ---

# R-squared

y=Y

y\_pred=predicted

r2 = r2\_score(y,y\_pred)

import seaborn as sns

# Calculate the residuals

residuals = y - y\_pred

# Convert to numpy arrays for plotting

y\_pred\_np = y\_pred.flatten()

residuals\_np = residuals.values.flatten()

# Plot residuals vs. predicted values

sns.scatterplot(x=y\_pred\_np, y=residuals\_np)

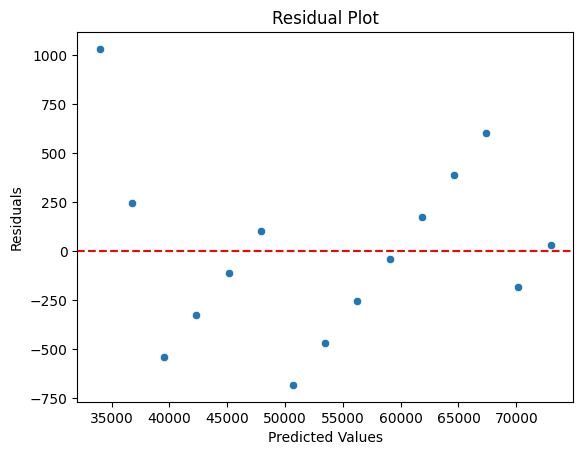
plt.axhline(y=0, color='r', linestyle='--')

plt.xlabel("Predicted Values")

plt.ylabel("Residuals")

plt.title("Residual Plot")

plt.show()



# Mean Absolute Error (MAE)

mae = mean\_absolute\_error(y, y\_pred)

# Mean Squared Error (MSE)

mse = mean\_squared\_error(y, y\_pred)

# Root Mean Squared Error (RMSE)

rmse = np.sqrt(mse)

print(f"R-squared (R²): {r2:.3f}")

print(f"Mean Absolute Error (MAE): {mae:.2f}")

print(f"Mean Squared Error (MSE): {mse:.2f}")

print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")

**R-squared (R²): 0.999**

**Mean Absolute Error (MAE): 345.40**

**Mean Squared Error (MSE): 191746.03**

**Root Mean Squared Error (RMSE): 437.89**