**Assignment 6**
**Name**: Tushar Patil
**Roll No**: 281035
**Batch**: A2

---

### Statement

This assignment focuses on applying Linear Regression to predict monthly temperature patterns using a suitable dataset. The main tasks include:

- Applying Linear Regression using Python libraries.
- Computing summary statistics for the dataset (mean, range, standard deviation, etc.).
- Evaluating the regression model using metrics such as MSE, MAE, and R².
- Visualizing feature distributions and the regression line.

---

### Objective

1. Gain familiarity with statistical summaries using Python.
2. Visualize data using histograms for deeper insights.
3. Preprocess and transform data for regression modeling.
4. Build, train, and evaluate a Linear Regression model for temperature forecasting.

---

### Resources Used

- **Software**: Visual Studio Code
- **Libraries**: Pandas, Matplotlib, Seaborn, Scikit-Learn

---

### Introduction to Pandas and Regression Modeling

Pandas is a versatile Python library for handling structured data efficiently. It supports operations such as reading files, cleaning data, and performing complex analytical tasks. In this assignment, Pandas helps prepare data for regression analysis, which is a key method for modeling relationships between dependent and independent variables.

---

### Key Capabilities of Pandas and Scikit-Learn

- Data loading from formats like CSV or Excel.
- Handling missing values and data transformation.
- Generating summary statistics.

- Training machine learning models like Linear Regression.
- Model evaluation using various metrics.

---

**Basic Python Functions and Methods Used**

1. pd.read_csv() – Load dataset from a CSV file.
2. DataFrame.describe() – View statistical summaries of numeric data.
3. hist() / sns.histplot() – Plot histograms to observe data distribution.
4. fillna() – Handle missing data using imputation.
5. train_test_split() – Divide the dataset for training and testing.
6. LinearRegression() – Apply linear regression model.
7. mean_squared_error(), mean_absolute_error(), r2_score() – Assess model accuracy.

---

**Methodology**

**1. Data Loading and Exploration**

- Imported monthly temperature dataset using Pandas.
- Checked for null values, data types, and column information.

**2. Preprocessing**

- Handled missing data using median imputation.
- Removed irrelevant features and retained the key predictors.

**3. Summary Statistics**

- Used describe() along with built-in NumPy functions to compute:
    - Minimum, maximum
    - Mean and median
    - Range and percentiles
    - Standard deviation and variance

**4. Data Visualization**

- Plotted histograms for each numerical column to analyze distributions.
- Identified patterns and skewness visually using Seaborn.

**5. Model Training**

- Split dataset using train_test_split() into 80% training and 20% testing.
- Built a LinearRegression() model and trained it on the training data.

- Predicted results on the test set.

**6. Evaluation**

- Measured accuracy using:
  - **MSE (Mean Squared Error)**
  - **MAE (Mean Absolute Error)**
  - **$R^2$ Score (Coefficient of Determination)**
- Visualized regression line with actual data points.
- Plotted residuals to assess model fit.

---

**Advantages**

- Easy manipulation of datasets using Pandas.
- Simple and interpretable model through Linear Regression.
- Quick visualization and pattern recognition using Seaborn.

---

**Disadvantages**

- Performance slows with large datasets.
- Linear Regression may not capture non-linear patterns well.

---

**Conclusion**

This assignment helped build a foundational understanding of regression analysis using Python. By working with Pandas and Scikit-Learn, we were able to preprocess real-world data, compute meaningful statistics, and predict temperatures effectively. Evaluation metrics validated our model, and visualizations provided intuitive insights into data behavior and model accuracy.