

## Assignment 2: Python Data Handling with Pandas and NumPy

### Objective

The aim of this assignment is to enhance Python programming skills by applying Pandas and NumPy for data manipulation and analysis. The assignment includes Python basics, data handling with Pandas, and numerical operations with NumPy.

---

### Section 1: Python Basics

#### Task 1: Python Functions and Lists

- Created a function that takes a list of numbers and returns a new list with each number squared.
- Used the function on a list of 10 integers and displayed both the original and squared lists.

#### Task 2: Dictionary Operations and Comprehensions

- Created a dictionary where keys are student names and values are their exam scores.
  - Implemented a function to filter and return students who scored above 50.
  - Used dictionary comprehension to create a new dictionary where each score is increased by 10%.
- 

### Section 2: Data Manipulation with Pandas

#### Task 1: Creating and Exploring a DataFrame

- Created a Pandas DataFrame using a dictionary containing names, ages, and cities.
- Displayed the first few rows of the DataFrame using `.head()`.
- Calculated the average age from the DataFrame using `.mean()`.

#### Task 2: Data Filtering and Aggregation

- Filtered the DataFrame to include only rows where age > 30 using boolean indexing.
- Added a new column AgeGroup to classify people as 'Adult' (age < 40) or 'Senior' (age ≥ 40).

- Grouped the data by AgeGroup and computed the average age for each group using `.groupby()` and `.mean()`.
- 

### **Section 3: Data Manipulation with NumPy**

#### **Task 1: NumPy Array Operations**

- Created a NumPy array with 12 random integers between 1 and 50 using `np.random.randint()`.
- Reshaped the array into a 3x4 matrix using `.reshape(3, 4)`.
- Calculated the maximum, minimum, and mean values using `np.max()`, `np.min()`, and `np.mean()`.

#### **Task 2: Array Indexing and Slicing**

- Created a NumPy array of shape (4, 3) using `np.arange(1, 13).reshape(4, 3)`.
  - Extracted the second row using array indexing.
  - Extracted a 2x2 submatrix from the top-left corner using slicing (`array[0:2, 0:2]`).
- 

#### **Tools Used**

- Python
  - Pandas
  - NumPy
  - Jupyter Notebook
- 

#### **Conclusion**

This assignment provided practical exposure to core Python concepts and hands-on experience with Pandas and NumPy. By working on real data structures and performing transformations, it strengthened the foundation for future work in data science and machine learning.