LAB ASSIGNMENT CHAPTER 2

1. Create a Pandas Series from a Python list [10, 20, 30, 40, 50] with custom indexes ['a','b','c','d','e'].

(a) Print the first 3 elements.

(b) Access the element at index 'c'.

2. Create a Series from a NumPy array of random integers between 1–100

   (size = 10).

(a) Find the maximum, minimum, and mean values.

(b) Apply a function to square each element.

3. Given a Series with values [5, np.nan, 8, np.nan, 12]:

(a) Check for missing values.

(b) Fill missing values with forward fill (ffill).

(c) Drop missing values.

4. Convert a Python dictionary

   data = {'Math': 85, 'Science': 90, 'English': 88}

   into a Series. Retrieve the value for 'Science'.

5. Create a DataFrame using a dictionary:

   data = { 'Name': ['Amit', 'Riya', 'John', 'Sara'],

   'Age': [25, 30, 22, 28],

   'Salary': [50000, 60000, 55000, 65000]. }

   (a) Select all rows where Age > 25.

   (b) Select rows where Salary is between 55,000 and 65,000.

6. Create a DataFrame:

   data = { 'Department': ['HR','IT','HR','IT','Finance'],

   'Employee': ['A','B','C','D','E'],

   'Salary': [40000, 50000, 42000, 55000, 60000]. }

(a) Find average salary per department.

(b) Count employees in each department.

(c) Sort employees by Salary in ascending order.

(d) Sort by Department then by Salary (descending).

7. Create a DataFrame with duplicate rows.

    (a) Identify the duplicates and Display only the duplicate rows.

    (b) Drop duplicates while:

        - Keeping the **first occurrence**
        - Keeping the **last occurrence**.
        - Removing **all duplicates**.

    (c) Drop duplicates based on specific columns .

    (d) Count the number of duplicate rows using duplicated().sum().

    (e) Extract only the unique values from a single column (e.g., Name).

8. Create a DataFrame containing NaN values.

    (a) Detect missing values

    (b) Count missing values per column.

    (c) Drop rows:

        - With **any NaN values**.
        - Where **all values are NaN**.
        - Where NaN appears in specific columns (like Age or Salary).

    (d) Fill missing values:

        - With a fixed value (e.g., 0 or "Unknown").
        - With the **mean** of the column.
        - Using **forward fill** and **backward fill**.
        - Using **linear interpolation**.

    (e) Compare the results of forward fill vs interpolation on the same dataset.