**Name:** Pavan B. Gavit

**Roll no:** 282005

**Batch:** B1

**Assignment 1**

**Statement:**

Q. In this assignment, we have to:

a) Read data from different formats (like CSV, XLS).

b) Find the shape of the data.

c) Identify missing values.

d) Determine the data type of each column.

e) Find the number of zeros in each column.

f) Perform indexing, selecting, and sorting of data.

g) Describe the attributes of the data and re-check data types.

h) Count unique values, examine the format of each column, and convert variable data types (e.g., from long to short, vice versa).

**Objective:**

1) This assignment aims to provide practical experience in data loading, exploration, and preprocessing using Pandas.

2) It familiarizes users with essential techniques for handling and manipulating dataframes.

3) Enhance our skills in data cleaning and preparation, improving our proficiency in data analysis.

**Resources used:**

1) Software used: Google Colab

2) Library used: Pandas

**Introduction to Pandas:**

1) Pandas is a powerful Python library for data manipulation and analysis.

2) It provides data structures like DataFrames for efficient data handling.

3) Pandas offers functions for reading data from various file formats, handling missing values, and performing data transformations.

Some basic functions that we used in the program:

1. `pd.read\_csv()`: Reads data from a CSV file.

2. `pd.read\_excel()`: Reads data from an Excel file.

3. `df.shape`: Returns the dimensions (rows, columns) of the DataFrame.

4. `df.isnull().sum()`: Counts the number of missing values in each column.

5. `df.dtypes`: Returns the data type of each column.

6. `(df == 0).sum()`: Counts the number of zeros in each column.

7. `df.loc[]`: Used for indexing and selecting data by label.

8. `df.iloc[]`: Used for indexing and selecting data by integer position.

9. `df.sort\_values()`: Sorts the DataFrame by one or more columns.

10. `df.describe()`: Generates descriptive statistics for numerical columns.

11. `df['column\_name'].unique()`: Returns unique values in a column.

12. `df['column\_name'].value\_counts()`: Counts the occurrences of each unique value.

13. `df['column\_name'].astype('new\_type')`: Converts the data type of a column.

**Methodology:**

**1. Data Loading:**

\* Read Data: Load data from CSV and Excel files using `pd.read\_csv()` and `pd.read\_excel()`.

**2. Data Exploration:**

\* Shape: Find the number of rows and columns using `df.shape`.

\* Missing Values: Identify missing values using `df.isnull().sum()`.

\* Data Types: Determine the data type of each column using `df.dtypes`.

\* Zeros: Count the number of zeros in each column using `(df == 0).sum()`.

**3. Data Manipulation:**

\* Indexing and Selection: Select specific rows and columns using `df.loc[]` and `df.iloc[]`.

\* Sorting: Sort the data based on column values using `df.sort\_values()`.

**4. Data Description and Transformation:**

\* Descriptive Statistics: Generate descriptive statistics using `df.describe()`.

\* Unique Values and Counts: Find unique values and their counts using `df['column\_name'].unique()` and `df['column\_name'].value\_counts()`.

\* Type Conversion: Convert data types using `df['column\_name'].astype('new\_type')`.

**Advantages:**

1. Pandas simplifies data loading and handling from various sources.

2. It provides efficient tools for data exploration and cleaning.

3. Pandas enables flexible data manipulation and transformation.

**Disadvantages:**

1. Pandas can be memory-intensive for very large datasets.

2. Certain operations might be computationally expensive.

**Conclusion:**

This assignment covered essential data handling techniques using Pandas. We learned to load data from different file formats, explore data characteristics, handle missing values, perform indexing and sorting, describe data attributes, count unique values, and convert data types. These skills are fundamental for any data analysis task and provide a solid foundation for more complex data manipulations.