**Name:** Aashish Suryawanshi

**Roll no:** 281021

**Batch:** A1

**Assignment 1**

**Statement:**

Q. Perform the following operations using R/Python on suitable data sets:  
a) Read data from different formats (like CSV, XLS)  
b) Find Shape of Data  
c) Find Missing Values  
d) Find Data Type of Each Column  
e) Finding Out Zeros  
f) Indexing and Selecting Data, Sort Data  
g) Describe Attributes of Data, Checking Data Types of Each Column  
h) Counting Unique Values of Data, Format of Each Column, Converting Variable Data Type (e.g., from long to short, vice versa)

**Objective:**

1. This assignment introduces the Pandas library and its core functionalities, focusing on reading different file formats such as CSV and Excel.
2. It also covers data cleaning and preprocessing techniques to improve data quality.
3. The goal is to enhance our ability to handle, manipulate, and analyze structured data effectively.

**Resources Used:**

1. Software: Jupyter Notebook
2. Library: Pandas

**Introduction to Pandas:**

1. Pandas is a powerful, open-source Python library designed for data analysis and manipulation.
2. It provides easy-to-use data structures and functions, making it an essential tool for handling structured data.
3. The library is built around two key data structures:
   * Series: A one-dimensional labeled array capable of storing various data types.
   * DataFrame: A two-dimensional labeled data structure with columns that can hold different types of data.
4. Pandas supports numerous data operations such as loading data from multiple file formats (CSV, Excel, SQL databases), cleaning, filtering, sorting, grouping, and performing statistical analyses.

**Key Functions Used in the Program:**

1. pd.read\_csv(): Loads data from a CSV file into a DataFrame.
2. shape: Returns the number of rows and columns in the dataset.
3. isnull().sum(): Identifies missing values within the dataset.
4. dtypes: Displays the data type of each column.
5. (df == 0).sum(): Counts the number of zero values in each column.
6. sort\_values(): Sorts the DataFrame based on a specified column in ascending or descending order.
7. describe(): Generates descriptive statistics such as count, mean, standard deviation, minimum, and maximum values for numerical columns.
8. unique(): Returns an array of unique values in a given column, useful for identifying distinct categories in categorical data.

**Methodology:**

1. Data Collection and Exploration:
   * Data Acquisition: Obtain a relevant dataset containing essential features.
   * Data Exploration: Load the dataset into a Pandas DataFrame and inspect its structure, including the number of rows, columns, data types, missing values, and inconsistencies.
2. Data Preprocessing:
   * Handling Missing Values: Identify and address missing values through imputation or removal.
   * Data Cleaning: Remove duplicates, rectify incorrect entries, and standardize formatting to ensure data consistency.
3. Feature Engineering:
   * Feature Selection: Identify the most relevant features using domain knowledge and statistical techniques.
   * Feature Encoding: Convert categorical data into numerical format using techniques like one-hot encoding or label encoding to facilitate analysis.

**Advantages of Pandas:**

1. User-friendly syntax, making it accessible for beginners and professionals alike.
2. Robust data structures like Series and DataFrame for efficient data handling.
3. Extensive functionality for data manipulation, allowing operations such as filtering, merging, and reshaping data.
4. Compatibility with other Python libraries such as NumPy and Matplotlib for advanced analysis and visualization.

**Disadvantages of Pandas:**

1. Handling very large datasets can lead to high memory consumption, impacting performance.
2. While highly integrated with Python, its interoperability with other programming languages is limited compared to some other data-processing tools.

**Conclusion:**

In conclusion, this assignment provided a foundational understanding of the Pandas library, a vital tool for data analysis and manipulation in Python. We explored essential functions for reading, organizing, and processing data efficiently. By applying Pandas to practical scenarios, we developed skills in handling structured data, cleaning and preprocessing information, and performing statistical analyses. These concepts serve as a stepping stone for more advanced data science and machine learning applications in the future**.**