ASSIGNMENT NO.1

Title: Perform the following operations using R/Python on suitable data sets:

a) read data from different formats (like csv, xls)

b) indexing and selecting data, sort data,

c) describe attributes of data, checking data types of each column,

d) counting unique values of data, format of each column, converting variable data type

(e.g. from long to short, vice versa),

e) identifying missing values and fill in the missing values

Software/Libraries Used:

- Python

- Pandas

Theory/Methodology:

Pandas is a powerful library in Python for data manipulation and analysis. It provides easy-to-use data structures and functions to perform various operations on structured data. Here's how we can utilize it for different tasks:

a) Reading Data from Different Formats:

- CSV: Using `pd.read_csv()`

- Excel: Using `pd.read_excel()`

b) Indexing and Selecting Data, Sorting Data:

- Indexing: Using `.loc[]` and `.iloc[]`

- Selection: Using column names or numeric indices

- Sorting: Using `.sort_values()`

c) Describing Attributes of Data, Checking Data Types:

- Describing: Using `.describe()`
- Data Types: Using `.dtypes`

d) Counting Unique Values, Converting Variable Data Types:

- Unique Values: Using `.value_counts()`
- Converting Data Types: Using `.astype()`

e) Identifying and Handling Missing Values:

- Identifying: Using `.isna()` or `.isnull()`
- Handling: Using `.fillna()` or `.dropna()`

Advantages/Application:

- Pandas offers a convenient way to handle and manipulate structured data, making it suitable for data preprocessing tasks in data science and machine learning projects.

- It provides a wide range of functionalities for data exploration, transformation, and cleaning, thus facilitating efficient data analysis workflows.

- Pandas integrates seamlessly with other Python libraries such as NumPy and Matplotlib, enabling comprehensive data analysis and visualization.

Limitations/Examples:

- Pandas may not be suitable for handling extremely large datasets due to its in-memory processing nature.

Working/Algorithm::

1. Read data from various formats using appropriate Pandas functions.

2. Perform indexing, selection, and sorting operations to extract relevant subsets of data.

3. Use descriptive statistics methods to understand the attributes of the data.

4. Count unique values and convert data types as needed.

5. Identify missing values and handle them by filling or removing them.

6. Utilize Pandas' functionalities to perform any additional data manipulation tasks required.

Conclusion:

Pandas is a versatile tool for data manipulation and exploration in Python, offering numerous functionalities to handle various data preprocessing tasks efficiently. By leveraging its capabilities, users can streamline the process of preparing data for analysis and gain valuable insights from their datasets. However, it's essential to be mindful of its limitations, particularly regarding scalability and performance with large datasets.