Assignment No:- 1

Problem Statement:

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 Library Package:

We'll use Python with the pandas library for data manipulation and analysis.

Theory:

i) Methodology:

Pandas is a powerful library for data manipulation and analysis in Python. It provides data structures like DataFrame and Series which are ideal for handling structured data. Here's a brief overview of the operations:

a) Reading Data: Pandas provides functions like `read_csv()` and `read_excel()` to read data from CSV and Excel files respectively.

b) Indexing and Selecting Data: You can use indexing and selection techniques like `loc[]` and `iloc[]` to select specific rows and columns from the DataFrame.

c) Sorting Data: The `sort_values()` function can be used to sort the data based on one or more columns.

d) Describing Attributes and Data Types: Functions like `info()`, `describe()`, and `dtypes` provide information about the DataFrame's attributes and data types of columns.

e) Counting Unique Values and Converting Data Types: `value_counts()` can be used to count unique values, and `astype()` can be used to convert data types.

f) Identifying and Filling Missing Values: Functions like `isna()`, `fillna()`, and `dropna()` help in identifying and filling missing values in the DataFrame.

ii) Advantages and Applications:

- Advantages: Pandas simplifies data manipulation and analysis tasks with its intuitive syntax and powerful functions. It integrates well with other libraries in the Python ecosystem such as NumPy and Matplotlib.
- Applications: Pandas is widely used in data preprocessing, data cleaning, exploratory data analysis, and feature engineering tasks in data science projects.

Limitations/Example:

Pandas may struggle with large datasets due to its in-memory processing nature. For very large datasets, alternative libraries like Dask or PySpark may be more suitable.

Working/Algorithm:

a) Reading Data:

Assume we have data stored in a CSV file named "employee_data.csv" containing information about employees such as their ID, name, department, and salary. We can use the `read_csv()` function from pandas to read this data into a DataFrame.

b) Indexing and Selecting Data:

After reading the data, we might want to select specific rows based on certain conditions, for example, employees belonging to the "Sales" department. We can use boolean indexing or the `query()` function to achieve this.

c) Sorting Data:

To analyze employee salaries effectively, we may need to sort the data based on the "Salary" column to identify the highest-paid employees. We can use the `sort_values()` function for this task.

d) Describing Attributes and Data Types:

Once we have the DataFrame, we can use methods like `info()` to get an overview of the data, `describe()` to obtain descriptive statistics, and `dtypes` attribute to check the data types of each column.

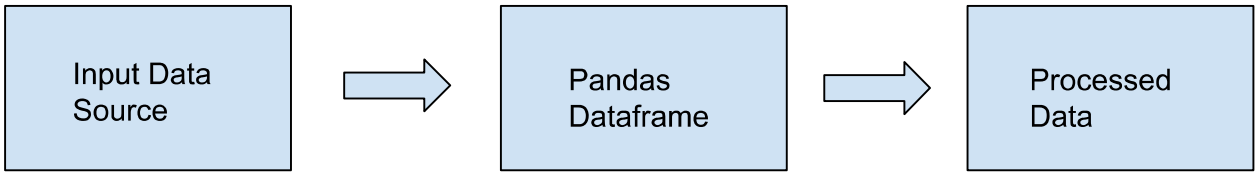
e) Counting Unique Values and Converting Data Types:

We may want to count the number of unique departments or employee IDs in the dataset. We can use the `value_counts()` function for this task. Additionally, if we need to convert data types, we can use the `astype()` method.

f) Identifying and Filling Missing Values:

During data analysis, we may encounter missing values in certain columns, such as "Salary" or "Department". We can use functions like `isna()` or `isnull()` to identify missing values, and then fill them using `fillna()` or `dropna()` as appropriate.

Diagram:



Conclusion:

By following these steps and utilizing pandas functionalities, we can effectively read, manipulate, and analyze data stored in various formats like Excel. Pandas provides a versatile toolkit for data handling in Python, making it a powerful choice for data scientists and analysts.