**** **Bansilal Ramnath Agarwal Charitable Trust’s**

**Vishwakarma Institute of Information Technology, Pune-48**

**(An Autonomous Institute affiliated to Savitribai Phule Pune University)**

**Department of Computer Science and Engineering (Artificial Intelligence)**

**LAB SUBMISSION**

**Data Science and Machine Learning**

**CAUA22201**

*Submitted by:*

*Avdhoot Nilesh Pimparkar,*

*PRN: 22210178,*

*Roll Number: 281054*

*Second Year*

*Semester II Academic Year 2023-24*

**Assignment: 1**

Aim: To perform the following operations using R/Python on suitable data sets:

1. read data from different formats (like .csv, .xls)
2. indexing and selecting data, sort data,
3. describe attributes of data, checking data types of each column,
4. counting unique values of data, format of each column, converting variable data type (e.g. from long to short, vice versa),
5. identifying missing values and fill in the missing values

Theory:

1. Reading Data from Different Formats:

Both R and Python offer libraries/packages to read data from various formats such as CSV, Excel, JSON, SQL databases, etc. In R, we can use functions like `read.csv()`, `read.table()`, or `readxl::read\_excel()` for Excel files. In Python, we can use libraries like Pandas (`pd.read\_csv()`, `pd.read\_excel()`), NumPy (`np.loadtxt()`), or openpyxl (`openpyxl.load\_workbook()`) for Excel files.

1. Indexing and Selecting Data, Sorting Data:

Indexing and selecting specific data from datasets allow you to focus on relevant information for analysis. In R, we can use indexing `[rows, columns]` or functions like `subset()` to filter data. Sorting can be done using `order()` or `dplyr` functions. In Python, Pandas offer powerful indexing and selection capabilities using `.loc[]`, `.iloc[]`, and sorting using `.sort\_values()`.

1. Describing Attributes of Data, Checking Data Types:

Understanding the structure of data is crucial. Descriptive statistics summarize the main characteristics of a dataset. R provides functions like `summary()`, `str()`, `class()`, and `sapply()` to describe data attributes and check data types. In Python, Pandas offer methods like `describe()`, `info()`, `dtypes`, and `apply()` to achieve similar tasks.

1. Counting Unique Values, Converting Variable Data Types:

Identifying unique values helps in understanding categorical variables, while converting data types is useful for compatibility and analysis. In R, you can use `table()`, `unique()`, `as.factor()`, `as.numeric()`, etc., to count unique values and convert data types. In Python Pandas, you can use `value\_counts()`, `unique()`, and `astype()` methods to achieve the same.

1. Identifying and Filling Missing Values:

Missing values can affect analysis results, so it is crucial to identify and handle them appropriately. Both R and Python provide functions to detect missing values (`is.na()` in R, `.isnull()` in Python). To fill missing values, R uses functions like `na.omit()`, `complete.cases()`, or `zoo::na.locf()`. In Python Pandas, you can use `.fillna()`, `.dropna()`, or `.interpolate()`.

These are foundational concepts for data manipulation and preprocessing in both R and Python. Understanding these operations is essential for effective data analysis and modelling. Practice and familiarity with relevant libraries/packages in each language enhances proficiency in data handling tasks.

Conclusion:

In this assignment, we were able to perform various operations on data like reading files, sorting data, finding null values and missing values, removing null values, etc.