DSS Lab Experiment Number: 4

AIM: Pandas Functions implementation on Data Frame

Pandas :

Importing data from CSV to DataFrame

We can also create a DataFrame by importing a CSV file. A CSV file is a text file with one record of data per line. The values within the record are separated using the "comma" character. Pandas provides a useful method, named read_csv() to read the contents of the CSV file into a DataFrame. For example, we can create a file named 'cities.csv' containing details of Indian cities. The CSV file is stored in the same directory that contains Python scripts. This file can be imported using:

Eg:

```
import pandas as pd
data =  pd.read_csv('file name') # file name or link where file is present
print(data)
```

Our aim is to load data and analyze it to draw conclusions. So, we can use any convenient method to load the data. In this tutorial, we are hard-coding the data of the DataFrame.

Inspecting data in DataFrame

Running the DataFrame using its name displays the entire table. In real-time, the datasets to analyze will have thousands of rows. For analyzing data, we need to inspect data from huge volumes of datasets. Pandas provide many useful functions to inspect only the data we need. We can use df.head(n) to get the first n rows or df.tail(n) to print the last n rows. For example, the below code prints the first 2 rows and last 1 row from the DataFrame.

1. Getting Statistical summary of records

We can get statistical summary (count, mean, standard deviation, min, max etc.) of the data using df.describe() function.

2. Sorting records

We can sort records by any column using df.sort_values() function.

3. Slicing records

It is possible to extract data of a particular column, by using the column name.

An interesting feature of Pandas library is to select data based on its row and column labels using iloc[0] function. Many times, we may need only few columns to analyze. We can also can select by index using loc['index_one']). For example, to select the second row, we can use df.iloc[1,:] . Let's say, we need to select second element of the second column. This can be done by using df.iloc[1,1] function. In this example, the function df.iloc[1,1]

4. Rename column

It is possible to use the df.rename() function to rename a column. The function takes the old column name and new column name as arguments.

5. Data Wrangling

Data Science involves the processing of data so that the data can work well with the data algorithms. Data Wrangling is the process of processing data, like merging, grouping and concatenating. The Pandas library provides useful functions like merge(), groupby() and concat() to support Data Wrangling tasks.

(Write Examples of each above functions and other functions in python)

Conclusion:

In above manner we have studied work with Pandas Library and application of its functions on Data Frame.