Y.C.C.E
DATE :

PRACTICAL 2

Aim :- Manipulating data using Pandas

Theory :- Pandas

Pandas is a newer package built on top of Numpy and provides an efficient implementation of a DataFrame. It is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring & manipulating data. It also consists of data structures and functions to perform efficient operations on data. It is well-suited for working with tabular data such as excel & SQL.

Installation of Pandas - Syntax `pip install pandas`

Importing Pandas - Syntax `import pandas`

* Advantages of Pandas

(i) Easy Data Handling
(ii) Data Analysis & Cleaning
(iii) Data Visualization
(iv) Performance
(v) Integration
(vi) User Friendly.

Y.C.C.E

DATE :

★ Functions Used in Manipulating Data Using Pandas

1) pd.read_csv :- It is used to load CSV data into a DataFrame which consists of records & fields.

2) df['column'].max() :- It is used to compute & display maximum value in dataset of column.

3) df.shape :- It is used to display no. of rows & columns.

4) df.head() :- Displays first five rows of dataset
5) df.tail() :- Displays last five rows of dataset
6) df.rows :- Displays all rows in dataset
7) df.columns :- Displays all column in dataset

8) df['column'].std() :- It calculates the standard deviation of values in particular column.

9) df.describe() :- It provides a summary of the statistical properties of numeric columns in dataframe.

10) df.set_index('column') :- It sets the column as the index of the DataFrame.

11) df.loc['index_value'] :- It is used to access rows in the DataFrame where the index_value exists.

Y.C.C.E

DATE :

★ Different Ways Of Creating DataFrame

1) Using CSV - Syntax :- pd.read_csv("file_name.csv")

2) Using Excel - Syntax :- pd.read_excel("file_name.xlsx","sheet_no")

3) Using Dictionary -

Syntax :-
```
data = { 'column 1' : [ 'values' ],
         'column 2' : [ 'values' ]
       }
df = pd.DataFrame(data)
df
```

4) Using Tuples

Syntax :-
```
data = [ ('values'), ('values') ]
df = pd.DataFrame(data = file_name, columns = [values])
df
```

5) Using List of Dictionaries

Syntax :-
```
data = [{'col1' : 'value', 'col2' : 'value', 'col3' : 'value'}]
df = pd DataFrame (data = file_name, colums = ['col1', 'col2']
```