AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

BE(E&C)	Data Science & Visualization Lab	
Experiment No.: 03	Importing and exporting CSV files using Pandas in Python and analyzing data (like shape, display of data in CSV file, checking missing value, and correlation) in CSV files.	Page: /5

Aim: Importing and exporting CSV files using Pandas in Python and analyzing data (like shape, display of data in CSV file, checking missing value, and correlation) in CSV files.

Software Used: Python 3.12, Jupyter Notebook.

Learning Objective

To learn how to import and export CSV files using the Pandas library in Python and analyze the data by exploring its shape, displaying the data, checking for missing values, and computing correlation.

Learning Outcomes:

After performing the experiment students will be able to-

Analyze the data by exploring its shape, displaying the data, checking for missing values, and computing correlation using Pandas in Python.

Theory:

1. Introduction to CSV Files and Pandas

CSV Files:

- CSV (Comma-Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database.
- Files in the CSV format are used to import/export large amounts of data between programs.

Pandas:

- Pandas is a powerful Python library used for data manipulation and analysis. It provides data structures like DataFrames to work efficiently with large datasets.

```
import pandas as pd
```

2. Importing CSV Files:

To read a CSV file using Pandas' read_csv function, you first need to import the Pandas library. Then, you can use the read_csv function by passing the path to the CSV file as an

argument. This function will return a DataFrame containing the data from the CSV file. This function helps us load the file from your local machine or any URL.

```
df = pd.read_csv("E:\Data Science and Visualization\DSV Practicals\diabetes.csv")
df
```

3. Read first and last 5 rows

The next step will be to show the top 5 rows and bottom 5 rows from our data collection using the pandas “head()” and “tail()” methods respectively. We can supply the number of rows we wish to show by passing the count as an argument to the "head()" function; for example, df.head(10) will now show 10 rows from our dataset.

```
df.head()
```

```
df.tail()
```

4. Pandas Shape Function/ Dimensionality of dataframe

The dimensions of our dataset may be viewed by using the "shape" function, which displays the dimensions in the format of (number of rows, number of columns).

```
Dimension = df.shape
print ("DIMENSION:", Dimension)
```

5. Pandas Size Function

The size property is used to get an int representing the number of elements in this object and Return the number of rows if Series. Otherwise, return the number of rows times the number of columns if DataFrame.

```
df.size
```

6. Index information of the DataFrame (label of the rows)

The index property returns the index information of the DataFrame. The index information contains the labels of the rows. If the rows has NOT named indexes, the

index property returns a RangeIndex object with the start, stop, and step values.

```
df.index
```

7. Pandas DataFrame columns Property

As the name suggests, "columns" get the name of all the features/columns in our data frame.

```
col = df.columns
print("Columns:\n", col)
```

8. Pandas DataFrame dtypes Property

The dtypes property returns data type of each column in the DataFrame.

```
datatype= df.dtypes
print(datatype)
```

9. Summary of a dataframe

The .info() method is a quick way to look at the data types, missing values, and data size of a DataFrame.

```
Summary = df.info()
print(Summary)
```

10. Description of the data in the DataFrame

The .describe() method prints the summary statistics of all numeric columns, such as count, mean, standard deviation, range, and quartiles of numeric columns.

```
desc = df.describe()
print(desc)
```

11. Description of memory usage in the DataFrame

This function return the memory usage of each column in bytes.

```
df.memory_usage()
```

12. Identification and Management of missing values

In order to check null values in Pandas DataFrame, we use isnull() function this function return dataframe of Boolean values which are True for NaN values.

```
df.isnull()
```

To count the number of missing values in each column of a DataFrame, use the isnull() method followed by the sum() method:

```
df.isnull().sum()
```

In order to check null values in Pandas Dataframe, we use notnull() function this function return dataframe of Boolean values which are False for NaN values.

```
df.notnull()
```

In order to drop a null values from a dataframe, we used dropna() function this function drop Rows/Columns of datasets with Null values in different ways.

```
df.dropna()
```

13. Correlation

Pandas dataframe.corr() is used to find the pairwise correlation of all columns in the Pandas Dataframe in Python. Any NaN values are automatically excluded. The corr() function is used to find the correlation among the columns in the Dataframe using 'Pearson' method**.**

```
df.corr()
```

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values,

darker colors are preferred.

```
correlation = df.corr()
sns.heatmap(correlation, annot=True, cmap='RdYlBu')
```

Conclusion:

_____

_____

_____

_____

Questions:

1. How would you identify and handle missing data in a CSV file using Pandas?
2. Explain with an example of a given CSV file, how would you calculate the correlation between two numeric columns?
3. Explain how you can display the first 10 rows of a CSV file after importing it into a Pandas DataFrame.
4. Write the syntax of heatmap with proper explanation.