

DSS Lab Experiment Number: 3

AIM: Study and Implementation of Pandas Library Functions in Python

NumPy :

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.
-

How to install pandas:

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, **pip**.

```
pip install pandas
```

After the pandas have been installed into the system, you need to import the library. This module is generally imported as:

```
import pandas as pd
```

Here, pd is referred to as an alias to the Pandas. However, it is not necessary to import the library using the alias, it just helps in writing less amount code every time a method or property is called.

Pandas generally provide two data structures for manipulating data, They are:

- **Series**
- **DataFrame**

Series: Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called indexes. Pandas Series is nothing but a column in an excel sheet. Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.

Creating a Series

In the real world, a Pandas Series will be created by loading the datasets from existing storage, storage can be SQL Database, CSV file, an Excel file. Pandas Series can be created from the lists, dictionary, and from a scalar value etc.

Eg:

```
import pandas as pd
import numpy as np
# Creating empty series
ser = pd.Series()
print(ser)
# simple array
data = np.array(['s', 's', 'g', 'm', 'c', 'e'])
ser = pd.Series(data)
print(ser)
\
```

Output:

```
Series([], dtype: float64)
```

```
0    s
```

```
1    s
```

```
2    g
```

```
3    m
```

```
4    c
```

```
5    e
```

```
dtype: object
```

DataFrame

Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

Creating a DataFrame:

In the real world, a Pandas DataFrame will be created by loading the datasets from existing storage, storage can be SQL Database, CSV file, an Excel file. Pandas DataFrame can be created from the lists, dictionary, and from a list of dictionaries, etc.

Eg:

```
import pandas as pd
```

```
data = {  
    "calories": [420, 380, 390],  
    "duration": [50, 40, 45]  
}
```

```
#load data into a DataFrame object:
```

```
df = pd.DataFrame(data)
```

```
print(df)
```

Output:

	calories	duration
0	420	50
1	380	40
2	390	45

Conclusion:

In above manner we have studied work with Pandas Library and its data types Functions