Workshop on Python (Day 4)

By Suriya G
Organized by Suresh Sir, UPNM

TABLE OF CONTENTS

What is Pandas?

- 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.

Installing Pandas

- Pip is a python package installer used to download & install any python packages:

pandas.py

```
# Basic syntax to install packages
pip install pandas
```

How to use Pandas in python?

- The recommendation convention to import is:

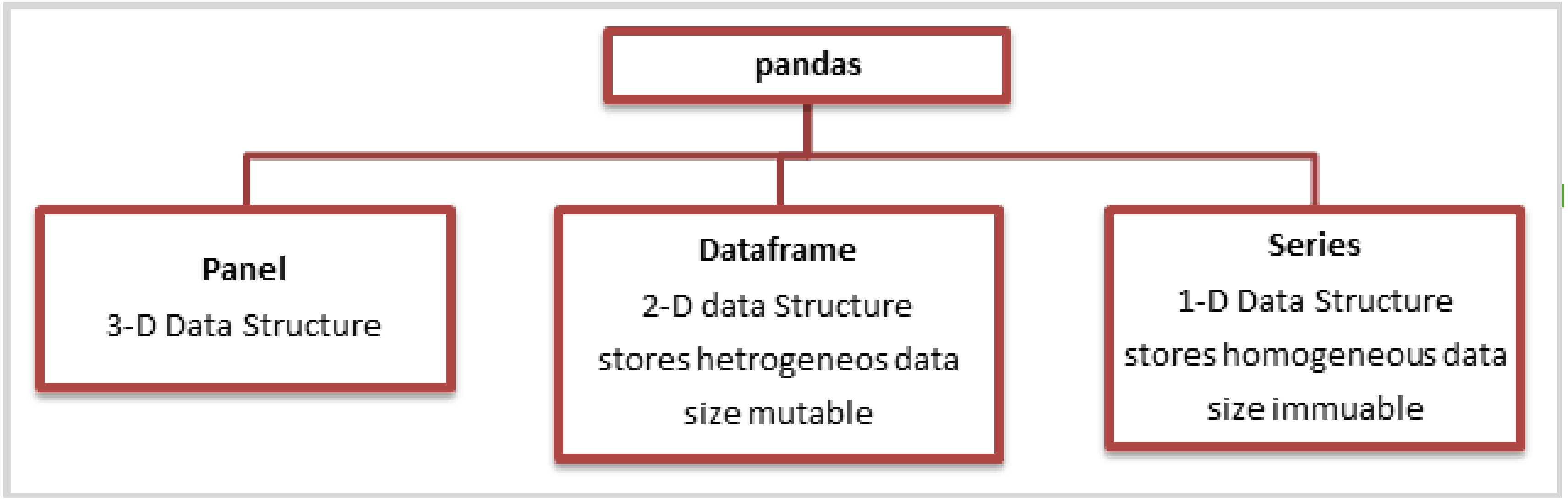
pandas.py

```
# Basic syntax to import packages
Import pandas as pd
```

Key Features of Pandas:

- Fast and Efficient
- Data alignment, missing values treatments can be done.
- Date type columns can be processed.
- Label-based slicing, indexing and subsetting of large data sets.
- High performance of merging and joining the data.
- Columns and rows can be deleted or inserted.

Pandas deal with 3 data structures:



Pandas data structures are MUTABLE!!

- All Pandas data structures are value mutable (can be changed)
- Series is size immutable.

Note − DataFrame is widely used and one of the most important data structures. Panel is used much less.

Pandas – Series:

- A Series is a **one-dimensional labeled array** capable of holding any data type **(integers, strings, floats, etc.).**
- **Real life Examples:**
  - Stock Prices - `pd.Series([100, 102, 105], index=["Mon", "Tue", "Wed"])`
  - Sensor Readings - `pd.Series([22.4, 23.1, 22.9], index=["10AM", "11AM", "12PM"])`
  - Student Marks - `pd.Series([85, 90, 78], index=["Math", "Sci", "Eng"])`

Pandas – Series:

series.py

```python
# Syntax of pandas series
pd.series(data,index,dtype,copy)
```

Data	Takes various forms like list, constants, float, integer
Index	It must be unique and hashable, same length as data
Dtype	It denotes data type of input. If None, DataType will be inferred.
Copy	Copy the data. By default it is False

Creating an empty series

- A basic series, which can be created in an Empty Series

pandas.py

```python
# Basic syntax to import packages
import pandas as pd
s = pd.Series()
print(s)
```

Creating Labels:

- A basic series, which can be created in an Empty Series

pandas.py

```python
# Basic syntax to import packages
import pandas as pd
a = [1, 7, 2]
myvar = pd.Series(a, index = ["x", "y", "z"])
print(myvar)
```

Pandas – DataFrame:

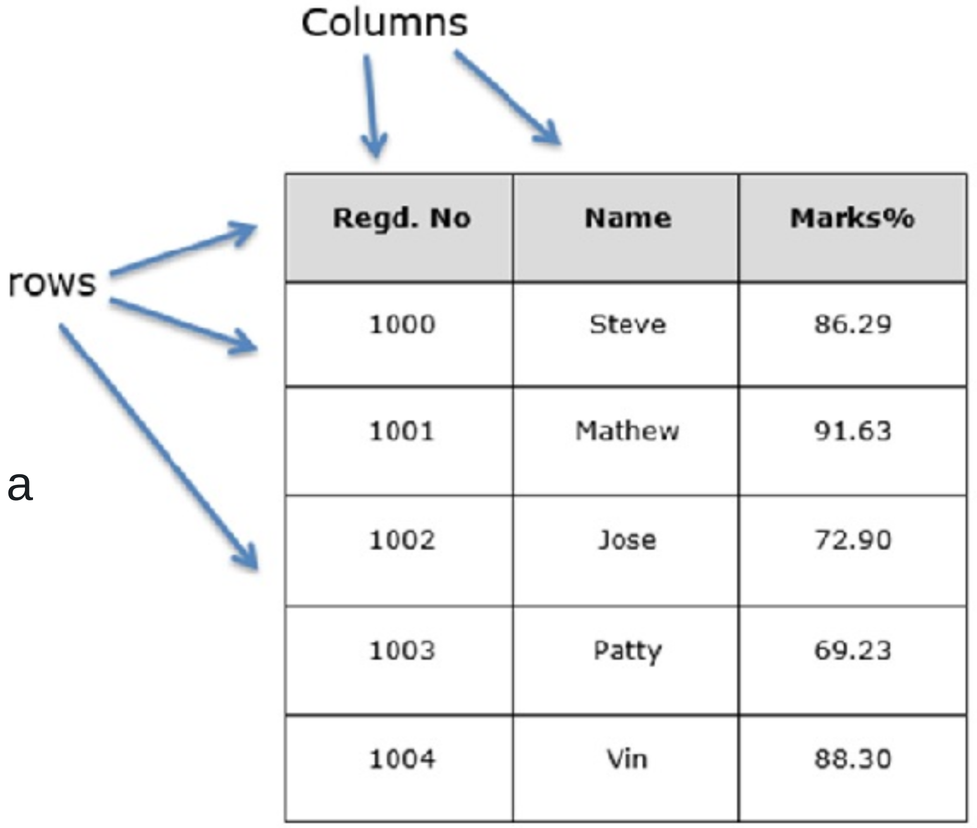
- A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.
- It's similar to a table or spreadsheet

Features of DataFrame

- Potentially columns are of different types
- Size – Mutable
- Labelled axes (rows and columns)
- Can Perform Arithmetic operations on rows and columns

Structure:

- Let us assume that we are creating a dataframe with student's data.
- You can think of it as an SQL table or a spreadsheet data representation.



Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

Pandas – DataFrame:

series.py

```
# Syntax of pandas series
pd.DataFrame(data,index,columns,dtype,copy)
```

Data	Takes various forms like list, constants, float, integer
Index	For the row labels
columns	For the column labels
Dtype	It denotes data type of input. If None, DataType will be inferred.
Copy	Copy the data. By default it is False

Creating a dataframe (Using Dict) :

dataframes.py

```
import pandas as pd
data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}
myvar = pd.DataFrame(data)
print(myvar)
```

Creating a dataframe (Using Dict) :

dataframes.py

```
import pandas as pd
data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}
myvar = pd.DataFrame(data,index=[‘day1’,’day2’,’day3’])
print(myvar)
```

Accessing custom rows:

dataframes.py

```
import pandas as pd
data = {
   "calories": [420, 380, 390],
   "duration": [50, 40, 45]
}
myvar = pd.DataFrame(data,index=[‘day1’,’day2’,’day3’])
print(df.loc["day2"])
```

Creating a dataframe (Using list) :

dataframes.py

```
import pandas as pd
data = [['Alex',10],['Bob',12],['Clarke',13]]
df = pd.DataFrame(data,columns=['Name','Age'])
print(df)
```

Exporting this Dataframe

- You can even export this into excel or csv file

dataframes.py

```
import pandas as pd
data = [['Alex',10],['Bob',12],['Clarke',13]]
df = pd.DataFrame(data,columns=['Name','Age’])
df.to_csv(“Exported_File.csv”)
```

Let's Play with own dataset:

Function	Definition (2 Points)	Example Code
1. read_csv()	Reads data from a CSV file into a DataFrame.	df = pd.read_csv("student_activities.csv")
2. head()	Returns the first 5 rows of the DataFrame.	df.head()
3. info()	Displays data types and non-null counts.	df.info()
4. describe()	Provides statistical summary for numerical columns.	df.describe()
5. shape	1 Gives the (rows, columns) of the DataFrame.	df.shape

6. columns	Lists all column names in the DataFrame.	df.columns
7. value_counts()	1 Counts occurrences of unique values in a column. 2 Helps understand category distributions.	df["Department"].value_counts()
8. groupby()	1 Groups data by one or more columns. 2 Useful for aggregation and summarization.	df.groupby("Department")["Score"].mean()
9. sort_values()	Can sort ascending or descending.	df.sort_values("Score", ascending=False)
10. loc[]	1 Accesses rows/columns by label/index. 2 Can be used for filtering with conditions.	df.loc[df["Score"] > 85]

11. isin()	1 Checks whether elements are in a list. 2 Handy for filtering multiple categories.	df[df["Department"].isin(["Electrical", "Civil"])]
12. isnull() + sum()	Detects missing values.	df.isnull().sum()
13. drop()	1 Removes rows or columns. 2 Axis = 0 (row), Axis = 1 (column).	df.drop("Attendance", axis=1)
14. fillna()	1 Fills missing values with specified value. 2 Often used for cleaning data.	df["Internship"].fillna("No", inplace=True)
15. apply()	Applies a function to a DataFrame or Series.	df["Score Category"] = df["Score"].apply(lambda x: "High" if x > 85 else "Low")