Getting started with Python-

Introduction to pandas Data Structures

Contents

Introduction to Pandas

pandas is a Python Package providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

- It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.
- It has a massive collection of many modules along with some unique features.

Source: pandas.pydata.org

Need for Pandas

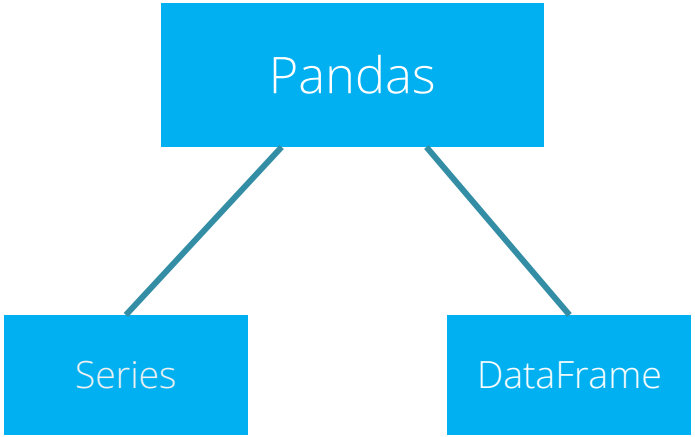
pandas makes data munging, preparing, analysis and modeling tasks easy and powerful with a few lines of code.

Some of the useful pandas techniques which gives it an edge over Python's built-in techniques:

- Creating DataFrames
- Loading data
- Crosstab
- Aggregating data
- Merging DataFrames
- Sorting DataFrames
- Plotting (Boxplot & Histogram)

Introduction to pandas Data Structures

Pandas have 2 key data structure:



- Import pandas library for data management and modelling tasks.

```
# import pandas in python
import pandas as pd
```

Series

- Series is a one dimensional labeled array object similar to list or column in a table.
- It is capable of holding any sort of data type (integers, floating point numbers, strings, dictionaries, etc).

Create a series with an arbitrary list:

```
s=pd.Series([8, 'data', 5.36, -23455788675342648, 'structures'])
s
```

```
0                    8
1                 data
2                 5.36
3   -23455788675342648
4           structures
dtype: object
```

Series

- You can explicitly define the index of Series, by default it starts from ‘Zero’.

Define index with `‘index=’` argument:

```
s=pd.Series([8, 'data', 5.36, -23455788675342648, 'structures'],
index=['A', 'B', 'C', 'D', 'E'])
s
```

```
A                     8
B                  data
C                  5.36
D    -23455788675342648
E            structures
dtype: object
```

Series

- Series can convert a dictionary as well.
- If index values are not explicitly defined using index argument, it uses the keys of the dictionary as its index.

Pass a Dictionary object to Series:

```
d={'01' : 'Jan', '02':'Feb','03':'Mar','04':'Apr'}
d
```

```
{'01': 'Jan', '02': 'Feb', '03': 'Mar', '04': 'Apr'}
```

```
months=pd.Series(d)
months
```

```
01    Jan
02    Feb
03    Mar
04    Apr
dtype: object
```

Series

Accessing elements of Series:

```
# Using Dictionary keys as its index
months['04']
```

```
'Apr'
```

```
# Using condition on value
months[months=='Jan']
```

```
01    Jan
dtype: object
```

DataFrame

- DataFrame is a 2-dimensional labeled data structure, similar to a spreadsheet or SQL table or a dictionary of Series objects.
- It can hold any type of data (integers, floating point numbers, strings, series, dictionaries, another dataframe, etc)
- Create a ‘basic_salary’ data with columns as ‘First_Name’, ‘Last_Name’, ‘Grade’, ‘Location’ and ‘ba’.

DataFrame

Pass a dictionary of lists to create a DataFrame

```
data={'First_Name':['Alan', 'Agatha', 'Rajesh', 'Ameet', 'Neha'],
     'Last_Name': ['Brown','Williams', 'Kolte', 'Mishra', 'Rao'],
     'Grade': ['GR1', 'GR2', 'GR1','GR2', 'GR1'],
     'Location': ['DELHI','MUMBAI', 'MUMBAI','DELHI', 'MUMBAI'],
     'ba':[17990, 12390, 19250, 14780, 19235]}
basic_salary = pd.DataFrame(data, columns=['First_Name',
'Last_Name', 'Grade', 'Location','ba’])
basic_salary
```

	First_Name	Last_Name	Grade	Location	ba
0	Alan	Brown	GR1	DELHI	17990
1	Agatha	Williams	GR2	MUMBAI	12390
2	Rajesh	Kolte	GR1	MUMBAI	19250
3	Ameet	Mishra	GR2	DELHI	14780
4	Neha	Rao	GR1	MUMBAI	19235

columns= argument allows us to give order of columns. By default they are

Default indices

alphabetically.

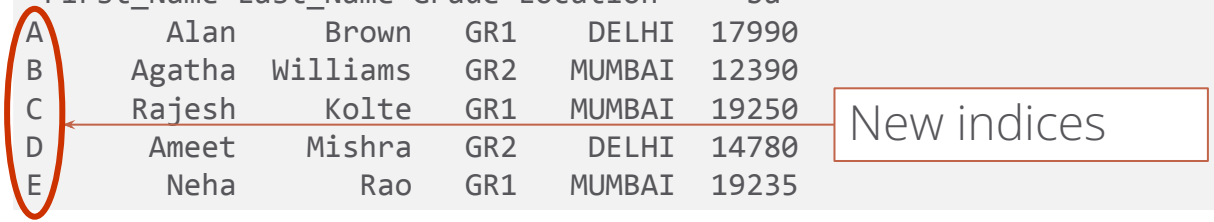
DataFrame

- Indexing (getting slices or chunks of data) in DataFrame is similar to indexing in Series
- Give index to all the columns of 'basic_salary'

Define index with '**index=**' argument

```
basic_salary = pd.DataFrame(data, index=['A','B','C','D','E'],
columns=['First_Name', 'Last_Name', 'Grade', 'Location', 'ba'])
basic_salary
```

```
  First_Name Last_Name Grade Location     ba
A       Alan     Brown   GR1    DELHI  17990
B     Agatha  Williams   GR2   MUMBAI  12390
C     Rajesh     Kolte   GR1   MUMBAI  19250
D      Ameet    Mishra   GR2    DELHI  14780
E       Neha       Rao   GR1   MUMBAI  19235
```



DataFrame

Accessing Columns

```
# Accessing columns using dictionary ‘key’ notation
basic_salary['First_Name']
# OR
basic_salary.First_Name
```

```
A      Alan
B    Agatha
C    Rajesh
D     Ameet
E      Neha
Name: First_Name, dtype: object
```

Both the syntax above returns the same output as basic_salary['First_Name']
Using ‘dot’ notation also, columns can be accessed.

DataFrame

Accessing Rows

```
# Accessing rows using row index label
basic_salary.loc[ 'B' , : ]
```

```
First_Name      Agatha
Last_Name     Williams
Grade              GR2
Location         MUMBAI
ba                12390
Name: B, dtype: object
```

Use integers instead of rows index : basic_salary.iloc[1,:]
Use integer range : basic_salary.iloc[1:3]

DataFrame

- Show records of employees from Location MUMBAI and rows from index B to E.

Slicing – using condition and row index

```
basic_salary[basic_salary.Location=='MUMBAI']
```

```
 First_Name Last_Name Grade Location     ba
B     Agatha  Williams   GR2   MUMBAI  12390
C     Rajesh     Kolte   GR1   MUMBAI  19250
E       Neha       Rao   GR1   MUMBAI  19235
```

```
# Slice along row indices
basic_salary.loc['B':'E']
```

```
 First_Name Last_Name Grade Location     ba
B     Agatha  Williams   GR2   MUMBAI  12390
C     Rajesh     Kolte   GR1   MUMBAI  19250
D      Ameet    Mishra   GR2    DELHI  14780
E       Neha       Rao   GR1   MUMBAI  19235
```

DataFrame

- Add a new column ‘ms’ to basic_salary.

Adding a new Column using dictionary syntax

```
# Add a new column ‘ms’
basic_salary['ms']=[16070,6630,14960,9300,15200]
basic_salary
```

```
 First_Name Last_Name Grade Location     ba     ms
A       Alan     Brown   GR1    DELHI  17990  16070
B     Agatha  Williams   GR2   MUMBAI  12390   6630
C     Rajesh     Kolte   GR1   MUMBAI  19250  14960
D      Ameet    Mishra   GR2    DELHI  14780   9300
E       Neha       Rao   GR1   MUMBAI  19235  15200
```

Quick Recap

Pandas	• pandas is a Python Package providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. • Import pandas in python : **`import pandas as pd`**
Series	• Series is a one dimensional labeled array object similar to list or column in a table. • **`pd.Series():`** Creates a series
DataFrame	• DataFrame is a 2-dimensional labeled data structure, which can hold any type of data. • **`pd.DataFrame(data,columns=[],index=[]):`** Creates a dataframe