DataFrames

July 22, 2021

1 Data Handling using Python Pandas (Part 2)

1.1 Quick Recap

PANDAS (**PAN**el **DA**ta) is a high level data manipulation tool used for analysing data.

- It is very easy to import and export data using Pandas Library which has a very rich set of functions.
- It is built upon packages like **NumPy** and **Matplotlib** (Package for Data Visualization)

1.1.1 Installing Pandas

Pandas does not come pre installed with the downloaded version of Python. We need to install Pandas module using the following command in the Command Prompt:-

pip install pandas

This command will help python install Pandas and all the necessary other files required for Pandas Module to run, like the Numpy and Matplotlib Modules. If these modules are already installed then the Prompt will show **Requirement Already Satisfied for these files**

1.1.2 Data Structures in Pandas

A **Data Structure** is a collection of data values and operations that can be applied to the data. It enables efficient storage, retrieval and modification of the data.

For example, **Python Lists, Dictionaries, Tuples, Numpy Arrays** are some common Data Structure we have already learned. Each of these Data Structures store more than one data value and have specific functionalities to work on them. Eg. - **Lists** have methods like **extend()** and **append()** - **Numpy Arrays** have methods like **zeros(), ones()** etc.

Python Pandas comes with 2 commonly used Data Structures:- 1. **Series** 2. **DataFrames**

Let us look at each one of them in Detail

1.2 DataFrames

A DataFrame is a **Two-dimensional** Labelled Data Structure like a Table. It contains rows and columns, and therefore has both a row and column index. ***Syntax:*** *pandas.DataFrame(data, [row index], [column index labels])*

[1]:
```
import pandas as pd

l2d = [[10,20,30,],
```

```
        [40,50,60],
        [70,80,90],
        [100,110,120]]
df = pd.DataFrame(l2d, columns = ['C1','C2','C3'])
df
```

[1]:

	C1	C2	C3
0	10	20	30
1	40	50	60
2	70	80	90
3	100	110	120

1.2.1 Properties of DataFrame

1. **Two Dimentional Data Stucture** - That is because it has rows and columns
2. **Labeled Indexes** - The rows and columns have indices.
3. **Heterogeneous Collection of Homogeneous Columns** - Each column will have similar data, however, the entire DataFrame can have multiple columns with Different Datatypes.
4. **Mutable Data** - Data Can be updated at any point in time.
5. **Flexible Size** - Rows and Columns can be added or removed after the creation of the DataFrame.
6. **Ordered** - It follows a specific order while displaying.

1.2.2 Creation of DataFrame

1. Empty DataFrame *Syntax: pandas.DataFrame()*

[2]:
```
import pandas as pd
df = pd.DataFrame()
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

2. Using a 2D List *Syntax: pandas.DataFrame(2D List)*

[6]:
```
import pandas as pd

L = [[10,    20,   30],
     [40,    50,   60],
     [100,  200,  300],
     [1000,3000, 6000]]
df_list = pd.DataFrame(L,columns = ['C1','C2','C3'], index = ['A','B','C','D'])
print(df_list)
```

```
     C1    C2    C3
A    10    20    30
B    40    50    60
```

```
C   100   200   300
D  1000  3000  6000
```

3. Using a 2D Numpy Array *Syntax: pandas.DataFrame(2D Numpy Array)*

[13]:
```
import pandas as pd
import numpy as np
array = np.arange(10,121,10)
array1 = array.reshape((4,3))
df_numpy = pd.DataFrame(array1,columns = ['N1','N2','N3'], index =␣
 ↪['R1','R2','R3','R4'])
#
print(array)
print(array1)
print(df_numpy)
```

```
[ 10  20  30  40  50  60  70  80  90 100 110 120]
[[ 10  20  30]
 [ 40  50  60]
 [ 70  80  90]
 [100 110 120]]
     N1   N2   N3
R1   10   20   30
R2   40   50   60
R3   70   80   90
R4  100  110  120
```

4. Using Dictionary of Lists *Syntax: pandas.DataFrame(Dict of List)*

[19]:
```
import pandas as pd

student = {
    'Name':['Ashish','Aklesh','Suman','Sushma'],
    'Age': [15,18,17,16],
    'Marks':[70,90,85,65],
    'Gender': ['M','M','F','F']
}

df_dict = pd.DataFrame(student,index = [1,2,3,4])
df_dict
```

[19]:
```
      Name  Age  Marks Gender
1  Ashish   15     70      M
2  Aklesh   18     90      M
3   Suman   17     85      F
4  Sushma   16     65      F
```

5. Using List of Dictionaries *Can be Considered like List of each record.* *Syntax: pandas.DataFrame(List of Dict)*

[20]:
```
import pandas as pd

student = [
    {'Name':'Ashish', 'Age':15,'Marks':70,'Gender':'M'},
    {'Name':'Aklesh', 'Age':18,'Marks':90,'Gender':'M'},
    {'Name':'Suman', 'Age':17,'Marks':85,'Gender':'F'},
    {'Name':'Sushma', 'Age':16,'Marks':65,'Gender':'F'}
]

df_list_dict = pd.DataFrame(student)
df_list_dict
```

[20]:

	Name	Age	Marks	Gender
0	Ashish	15	70	M
1	Aklesh	18	90	M
2	Suman	17	85	F
3	Sushma	16	65	F

6. Using Dictionary of Series *Syntax: pandas.DataFrame(Dict of Series)*

[22]:
```
import pandas as pd

Name_df = pd.Series(['Ashish','Aklesh','Suman','Sushma'])
Age_df = pd.Series([15,18,17,16])
Marks_df = pd.Series([70,90,85,65])
Gender_df = pd.Series( ['M','M','F','F'])

student = {'Name': Name_df, 'Age':Age_df,
           'Marks':Marks_df,'Gender':Gender_df}

df_series = pd.DataFrame(student)
df_series
```

[22]:

	Name	Age	Marks	Gender
0	Ashish	15	70	M
1	Aklesh	18	90	M
2	Suman	17	85	F
3	Sushma	16	65	F

Let us Practice

#Assignment 1: Create a DataFrame Players with columns Name, Country, Role, Batting Hand. Include the details of 5 players.

#Assignment 2: Create a DataFrame Food Menu with the coulmns Title, Veg/Non Veg, Price. Enter the Food Details of 5 Food Items.

[2]:
```
import pandas as pd

emp = {
    'Name':['Ramesh','Suresh','Sumukhi','Tanmay','Biswa'],
    'Department': ['Logistics','Logistics','Creative','Creative','Editorial'],
```

```
    'Salary': [50000,60000,45000,60000,50000],
    'Location': ['Delhi','Mumbai','Mumbai','Hyderabad','Kolkata']
      }
Employee = pd.DataFrame(emp)
Employee
```

[2]:

	Name	Department	Salary	Location
0	Ramesh	Logistics	50000	Delhi
1	Suresh	Logistics	60000	Mumbai
2	Sumukhi	Creative	45000	Mumbai
3	Tanmay	Creative	60000	Hyderabad
4	Biswa	Editorial	50000	Kolkata

1.2.3 Operations on Rows and Columns

1. Adding Columns to a DataFrame

[3]:
```
import pandas as pd
bonus = [500,500,500,500,500]
Employee['Bonus'] = bonus
Employee
```

[3]:

	Name	Department	Salary	Location	Bonus
0	Ramesh	Logistics	50000	Delhi	500
1	Suresh	Logistics	60000	Mumbai	500
2	Sumukhi	Creative	45000	Mumbai	500
3	Tanmay	Creative	60000	Hyderabad	500
4	Biswa	Editorial	50000	Kolkata	500

[4]:
```
Employee['Gross Salary'] = Employee['Salary'] + Employee['Bonus']
Employee
```

[4]:

	Name	Department	Salary	Location	Bonus	Gross Salary
0	Ramesh	Logistics	50000	Delhi	500	50500
1	Suresh	Logistics	60000	Mumbai	500	60500
2	Sumukhi	Creative	45000	Mumbai	500	45500
3	Tanmay	Creative	60000	Hyderabad	500	60500
4	Biswa	Editorial	50000	Kolkata	500	50500

[7]:
```
Employee['Gross Salary'] = Employee['Gross Salary'] + 10000
Employee
```

[7]:

	Name	Department	Salary	Location	Gross Salary
0	Ramesh	Logistics	50000	Delhi	70500
1	Suresh	Logistics	60000	Mumbai	80500
2	Sumukhi	Creative	45000	Mumbai	65500
3	Tanmay	Creative	60000	Hyderabad	80500
4	Biswa	Editorial	50000	Kolkata	70500

#Assignment 3: Create a DataFrame with the columns Roll, Name, Marks(40). Take 5 records of students. After the DataFrame is created, add a new column Percentage and Calculate the

Percentage using the formula (Marks * 100/40)

2. Adding Rows to a DataFrame

[12]:
```
Employee.loc[len(Employee)] = ['Kenny', 'Editorial', 55000,'Mangalore', 70000]
```

[13]:
```
Employee
```

[13]:

	Name	Department	Salary	Location	Gross Salary
0	Ramesh	Logistics	50000	Delhi	70500
1	Suresh	Logistics	60000	Mumbai	80500
2	Sumukhi	Creative	45000	Mumbai	65500
3	Tanmay	Creative	60000	Hyderabad	80500
4	Biswa	Editorial	50000	Kolkata	70500
5	Kenny	Editorial	55000	Mangalore	70000

3. Deleting Columns from a DataFrame Function used to delete the column is **drop()** `SYNTAX: df.drop(Name of Column, axis_value[,inplace --> False])`

[6]:
```
#Method 1: Asigning the New Dataframe to the same Variable
import pandas as pd

Employee = Employee.drop('Bonus',axis = 1)
Employee
```

[6]:

	Name	Department	Salary	Location	Gross Salary
0	Ramesh	Logistics	50000	Delhi	60500
1	Suresh	Logistics	60000	Mumbai	70500
2	Sumukhi	Creative	45000	Mumbai	55500
3	Tanmay	Creative	60000	Hyderabad	70500
4	Biswa	Editorial	50000	Kolkata	60500

[42]:
```
#Method 2: Using inplace parameter
Employee.drop('Gross Salary', axis = 1, inplace = True)
Employee
```

[42]:

	Name	Department	Salary	Location
0	Ramesh	Logistics	50000	Delhi
1	Suresh	Logistics	60000	Mumbai
2	Sumukhi	Creative	45000	Mumbai
3	Tanmay	Creative	60000	Hyderabad
4	Biswa	Editorial	50000	Kolkata

4. Deleting Rows from a DataFrame

[14]:
```
Employee.drop(5,inplace = True)
Employee
```

[14]:

	Name	Department	Salary	Location	Gross Salary
0	Ramesh	Logistics	50000	Delhi	70500
1	Suresh	Logistics	60000	Mumbai	80500

```
2  Sumukhi   Creative   45000     Mumbai         65500
3   Tanmay   Creative   60000  Hyderabad         80500
4    Biswa  Editorial   50000    Kolkata         70500
```

To drop the row with the Department Value as Editorial

[ ]:
```
idx = Employee.loc[Employee.Department == 'Creative'].index
# Provides the positional index values of rows which satisfy the condition -->
 ↪(2,3)
Employee.drop(idx, inplace = True)
```

[20]:
```
Employee
```

[20]:
```
     Name Department  Salary Location  Gross Salary
0  Ramesh  Logistics   50000    Delhi         70500
1  Suresh  Logistics   60000   Mumbai         80500
4   Biswa  Editorial   50000  Kolkata         70500
```

#Assignment 4: Create 5 new rows for the DataFrame created in Assignment 1. Then Remove All the rows where the country is 'Australia'.

[24]:
```
import pandas as pd

player = {'Name': ['Rohit', 'Smith','Warner', 'Sachin', 'Morgan'],
          'Country': ['India', 'Australia','Australia','India','England'],
          'Role:': ['Bat','Bat','Bat','Bat','Bat'],
          'Batting Hand': ['R','R','L','R','L']}
Player_df = pd.DataFrame(player)
Player_df.loc[len(Player_df)] = ['Russel','West Indies','Bat','R']
Player_df.loc[len(Player_df)] = ['Dhoni','India','WK','R']
Player_df.loc[len(Player_df)] = ['Paine','Australia','WK','R']
Player_df.loc[len(Player_df)] = ['Archer','England','Bowler','R']
Player_df.loc[len(Player_df)] = ['Stark','Australia','Bowler','L']

idx = Player_df.loc[Player_df.Country == "Australia"].index
Player_df.drop(idx, axis = 0, inplace = True)
Player_df
```

1.2.4 Accessing Elements from a DataFrame

We have to use loc / iloc

[8]:
```
Employee
```

[8]:
```
     Name Department  Salary   Location  Gross Salary
0  Ramesh  Logistics   50000      Delhi         70500
1  Suresh  Logistics   60000     Mumbai         80500
2  Sumukhi   Creative   45000     Mumbai         65500
3   Tanmay   Creative   60000  Hyderabad         80500
4    Biswa  Editorial   50000    Kolkata         70500
```

[9]:
```
Employee.loc[Employee['Gross Salary'] >60000, : ]
```

[9]:
```
      Name Department  Salary   Location  Gross Salary
0   Ramesh  Logistics   50000      Delhi         70500
1   Suresh  Logistics   60000     Mumbai         80500
2  Sumukhi   Creative   45000     Mumbai         65500
3   Tanmay   Creative   60000  Hyderabad         80500
4    Biswa  Editorial   50000    Kolkata         70500
```

[10]:
```
Employee.loc[Employee.Location == 'Mumbai',['Name','Salary','Location']]
```

[10]:
```
      Name  Salary Location
1   Suresh   60000   Mumbai
2  Sumukhi   45000   Mumbai
```

[11]:
```
Employee.loc[Employee.Department =='Logistics', ['Name','Salary']]
```

[11]:
```
     Name  Salary
0  Ramesh   50000
1  Suresh   60000
```

1.2.5 Renaming Rows and Columns

Syntax: *DataFrame.rename(dictionary{old:new}, axis[, inplace])*

[32]:
```
import pandas as pd

player = {'Name': ['Rohit', 'Smith','Warner', 'Sachin', 'Morgan'],
          'Country': ['India', 'Australia','Australia','India','England'],
          'Role:': ['Bat','Bat','Bat','Bat','Bat'],
          'Batting Hand': ['R','R','L','R','L']}
Player_df = pd.DataFrame(player)
Player_df.loc[len(Player_df)] = ['Russel','West Indies','Bat','R']
Player_df.loc[len(Player_df)] = ['Dhoni','India','WK','R']
Player_df.loc[len(Player_df)] = ['Paine','Australia','WK','R']
Player_df.loc[len(Player_df)] = ['Archer','England','Bowler','R']
Player_df.loc[len(Player_df)] = ['Stark','Australia','Bowler','L']

Player_df.rename({0:'Open 1',1: 'Open 2', 2: '1 Down', 3: '2 Down', 4: '3␣
 ↪Down', 5: '4 Down',
                  6: '5 Down',7: '6 Down',8: '7 Down', 9:'8 Down'}, axis = 0,␣
 ↪inplace = True)

Player_df.rename({'Name':'Player Name','Role:':'Role'}, axis = 1, inplace =␣
 ↪True)
print(Player_df)
```

```
       Player Name      Country    Role Batting Hand
Open 1       Rohit        India     Bat            R
Open 2       Smith    Australia     Bat            R
1 Down      Warner    Australia     Bat            L
2 Down      Sachin        India     Bat            R
```

```
3 Down      Morgan      England     Bat            L
4 Down      Russel  West Indies     Bat            R
5 Down       Dhoni        India      WK            R
6 Down       Paine    Australia      WK            R
7 Down      Archer      England  Bowler            R
8 Down       Stark    Australia  Bowler            L
```

1.2.6 Merging of DataFrames

1. Append()
2. Merge()

1. Append ***SYNTAX: df1.append(df2)***

[37]:
```
import pandas as pd
import numpy as np
df1 = pd.DataFrame(np.arange(10,121,10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
                   index = ['A','B','C','D'])

df2 = pd.DataFrame(np.arange(11,122,10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
                   index = ['E','F','G','H'])
print("Original DF 1")
print(df1)
print("Original DF 2")
print(df2)

#print("Appending DF2 to DF1")
#df1 = df1.append(df2)
#print(df1)

print('Appending DF1 to DF2')
df2 = df2.append(df1)
print(df2)
```

```
Original DF 1
    C1   C2   C3
A   10   20   30
B   40   50   60
C   70   80   90
D  100  110  120
Original DF 2
    C1   C2   C3
E   11   21   31
F   41   51   61
G   71   81   91
H  101  111  121
```

```
Appending DF1 to DF2
    C1   C2   C3
E   11   21   31
F   41   51   61
G   71   81   91
H  101  111  121
A   10   20   30
B   40   50   60
C   70   80   90
D  100  110  120
```

[44]:
```
import pandas as pd
import numpy as np
df1 = pd.DataFrame(np.arange(10,121,10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
                   index = ['A','B','C','D'])

df2 = pd.DataFrame(np.arange(121,10,-10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
                   index = ['A','F','C','H'])
print("Original DF 1")
print(df1)
print("Original DF 2")
print(df2)


df3 = df1.append(df2, sort = True )
print(df3)
```

```
Original DF 1
    C1   C2   C3
A   10   20   30
B   40   50   60
C   70   80   90
D  100  110  120
Original DF 2
    C1   C2   C3
A  121  111  101
F   91   81   71
C   61   51   41
H   31   21   11
    C1   C2   C3
A   10   20   30
B   40   50   60
C   70   80   90
D  100  110  120
A  121  111  101
```

```
F   91   81   71
C   61   51   41
H   31   21   11
```

1.2.7 Properties / Attributes of Dataframe

1. index
2. columns
3. values
4. dtypes
5. shape
6. size
7. Transpose
8. head()
9. tail()

[61]:
```
import pandas as pd
import numpy as np
df1 = pd.DataFrame(np.arange(10,121,10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
                   index = ['A','B','C','D'])

print(df1)
print('To Display the row indexes of the DF we write df.index')
print(df1.index)
print('To display the column indexes of the DF we write df.columns')
print(df1.columns)
print('To display only the values inside the DF we use df.values')
print(df1.values)
print('To display the values of a coulmn we use df.column.values or df.loc[:
 ↪,"coulum"].values')
print(df1.loc[:,['C1','C2']].values)
print('To display the values of specific rows we use df.loc[[r1,r2,...],:]')
print(df1.loc[['A','D'],:].values)
print('To display the datatypes of all columns')
print(df1.dtypes)
print("To display the first 2 rows we use df.head(n). By Default n = 5")
print(df1.head(2))
print("To display the last 3 rows we use df.tail(n). By Default n = 5")
print(df1.tail(3))
print('To find the shape of the dataframe')
print(df1.shape)
print('To find the size of the DataFrame')
print(df1.size)
```

```
    C1   C2   C3
A   10   20   30
B   40   50   60
```

```
C   70   80   90
D  100  110  120
To Display the row indexes of the DF we write df.index
Index(['A', 'B', 'C', 'D'], dtype='object')
To display the column indexes of the DF we write df.columns
Index(['C1', 'C2', 'C3'], dtype='object')
To display only the values inside the DF we use df.values
[[ 10  20  30]
 [ 40  50  60]
 [ 70  80  90]
 [100 110 120]]
To display the values of a coulmn we use df.column.values or
df.loc[:,"coulum"].values
[[ 10  20]
 [ 40  50]
 [ 70  80]
 [100 110]]
To display the values of specific rows we use df.loc[[r1,r2,...],:]
[[ 10  20  30]
 [100 110 120]]
To display the datatypes of all columns
C1    int32
C2    int32
C3    int32
dtype: object
To display the first 2 rows we use df.head(n). By Default n = 5
   C1  C2  C3
A  10  20  30
B  40  50  60
To display the last 3 rows we use df.tail(n). By Default n = 5
    C1   C2   C3
B   40   50   60
C   70   80   90
D  100  110  120
To find the shape of the dataframe
(4, 3)
To fond the size of the DataFrame
12
```

[64]:
```
import pandas as pd
import numpy as np
df1 = pd.DataFrame(np.arange(10,121,10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
                   index = ['A','B','C','D'])

df2 = pd.DataFrame(np.arange(121,10,-10).reshape((4,3)), columns =␣
 ↪['C1','C2','C3'],
```

```
                  index = ['E','F','G','H'])

df3 = df1.append(df2)

print(df3)
print()
print("Transpose of a DataFrame we use df.T")
print(df3.T)
```

```
    C1   C2   C3
A   10   20   30
B   40   50   60
C   70   80   90
D  100  110  120
E  121  111  101
F   91   81   71
G   61   51   41
H   31   21   11

Transpose of a DataFrame we use df.T
     A   B   C    D    E   F   G   H
C1  10  40  70  100  121  91  61  31
C2  20  50  80  110  111  81  51  21
C3  30  60  90  120  101  71  41  11
```

1.2.8 Importing Data from CSV file to a DataFrame

[65]:
```
import pandas as pd

employee = pd.read_csv('Employee.csv')
employee
```

[65]:

	EmpID	Ename	Department	Salary	City
0	E101	Arkopal	CS	50000	Kolkata
1	E102	Abhishek	History	50000	Gorakhpur
2	E103	Madav	Geography	60000	Ranchi
3	E104	Sarwan	Chemistry	70000	Jaipur
4	E105	Dr. Dubey	Principal	90000	Allahabad

[ ]:
```

```