

Pandas- DataFrame- Basics

Ex.No.

Date:

Aim:

To install Pandas package and do the basic DataFrame operations

Description:

1. Declaring empty DataFrame
 2. Declare and print the DataFrame series
 3. Do the basic functions such as extracting one column and one row, adding one column and
- FOW

```
import pandas as pd
```

```
#Empty DataFrame
```

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

```
print (df)
```

```
emp = pd.Series(['Parker', 'John', 'Smith', 'William'])
```

```
id=pd.Series([102, 107, 109, 114])
```

```
frame = ( 'Emp': emp, 'ID': id )
```

```
result = pd.DataFrame(frame)
```

```
print("\nSeries to Data frame\n")
```

```
print(result)
```

```
print("\n Extracting one column :\n")
```

```
print(result['Emp'])
```

```
print("\nAdding new column:\n")
```

```
result['Age']=pd.Series([35,24,40,38])
```

```
print(result)
```

```
prin("\nDeletingone colnmn:\n"
```

```
del result['Age']
```

```
print(result)
```

```
print("\nExtracting the third row:\n")
```

```
print(result.loc[2])
```

```
print("\nSlice rows:\n",result[1:3])
```

```
d2 = pd.DataFrame([[ 'Dale', 123], [ 'Mark', 143]], columns = ['Emp','ID'])
```

```
print("\nAdding new row values:\n",result.append(d2))
```

```
print("\nDeleting particular row:\n",result.drop[1])
```

Output:

Empty DataFrame

Columns: []

Index: []

Series to Data frame

Emp ID

Parker 102

1 John 107

2 Smith 109

3 William 114

Extracting one column:

Parker

1 John

2 Smith

3 William

Name: Emp, dtype: object

Adding new column:

```

Emp ID Age
0 Parker 102 35 1
John 107 242
Smith 109 40 3
William 114 38

```

Deleting one column".

```

Emp ID
Parker 102
1 John 107
2 Smith 109
3 William 114

```

Extracting the third row:

```

Emp Smith
ID 109
Name: 2, dtype: object

```

Slice rows:

```

Emp ID
1 John 107
2 Smith 109

```

Adding new row values:

```

Emp ID
Parker 102
1 John 107
2 Smith 109
3 William 114
Dale 123
1 Mark 143

```

Oeleting particular row:

	Emp ID
0	Parker 102
2	Smith 109
3	William 114

Pandas - DataFrame - Functions

Ex.No.

Date:

Aim:

To perform Pandas DataFrame functions

Description:

1. Apply sum and square root functions through Numpy
2. Perform min and max operations through aggregation function
3. Add new column using assign function
4. Do sort and merge functions

Program:

```
import pandas as pd
import numpy as np

#DataFrame.Apply function
print("Data Frame.apply function:\n")
info = pd.DataFrame([(2, 7)]*3, columns=['P', 'Q'])
print("\nOriginal DataFrame:\n",info)

print("\nSquare root of DataFrame:\n",info.apply(np.sqrt))
print{ "\nSum of each Column:\n",info.napply(np.sum, axis=0)}
print{"\nSum of each row:\n",info.apply(np.sum, axis=1)}

info=pd.DataFrame([(1,5,7),(2,7,8),(3,6,9)],columns=['X','Y','Z'])
print("\nOriginal DataFrame:\n",info)
print{ "\nMinimum and Maximum of each column: \n"}
print(info.agg(['min','max']))

#DataFrame.Assign function
print("\nDataFrame.Assign function:\n")
d2 = pd.DataFrame([('Dale', 123), ('Mark', 143)], columns = ('Emp','ID'))
print("\nOriginal DataFrame:\n",d2)
d2['Age']={35,40}
```

```

print("\nAdding new column:\n",d2)

d=d2.assign(Sex=['Male','Male'])

print("\nAdding new column:\n",d)


#DataFrame.Sort function

print("\nDataFrame.Sortfunction:\n")

info=pd.DataFrame(np.random.randn(5,2),index=[3,2,0,4,1],columns = ('col3','col4'))

print(info)

info2=info.sort_index()

print("\nSort index:\n",info2)

#info =

pd.DataFrame({'col1':[7,1,8,3], 'col2':[8,12,4,9]}) info3

= info.sort_values(by='col3')

print("\nSort Values:\n",info3)


#DataFrame.Merge function

print("\nDataFrame.Merge function:\n")

left = pd.DataFrame((

    'id':[1,2,3,4,5],

    'Name' ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],

    'sub':['sub1','sub2','sub4','sub6','sub5']))

right = pd.DataFrame((

    'id':[1,2,3,4,5],

    'Name' ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],

    'sub':['sub2','sub4','sub3','sub6','sub5']))

print(left)

print(right)

print(pd.merge(left,right,on='id'))

```

Output:

DataFrame.apply
function:

Original DataFrame:

P Q

0 2 7

1 2 7

2 2 7

Square root of DataFrame:

P Q

0 1.414214 2.645751

1 1.414214 2.645751

2 1.414214 2.645751

Sum of each column:

Q 21

dtype: int64

Sum of each row:

0 9

1 9

2 9

dtype: int64

Original DataFrame:

0 1 5 7

1 2 7 8

2 3 6 9

Minimum and Maximum of each column:

min 1 5 7

max 3 7 9

DataFrame.Assign function:

Original DataFrame:

Emp ID

0 Dale 123

1 Mark 143

Adding new column:

Emp ID Age

0 Dale 123 25

1 Mark 143 40

Adding new column:

Emp ID Age Sex

Dale 123 35 Male

1 Mark 143 40 Male

DataFraue.Sol t function:

col3 col4

3 -0.799987 -0.987387

2 -0.590037 0.838657

0 0.084095 -0.370267

4 0.375498 1.134862

1 -1.004183 0.314957

Son index:

col3 col4

0 0.084095 -0.370267

1 -1.004183 0.314957

2 -0.590037 0.838657

3 -0.799987 -0.987387

4 0.375498 1.134862

Soit Values:

col3 col4

1 -1.004183 0.314957

3 -0.799987 -0.987387

2 -0.590037 0.838657

0 0.08409s —0.370267

4 0.37S49B 1.134862

DataFrame.Merge function:

E'
 N suS
, d A:

bS
 i e'

0 1 Billy sub2

BB

2 3 :n sub4

i ?l El' x b x Name y sub y

1 Alex sub1 Billy sub2

e n b b It b5

Pandas Series

Ex.No.

Date:

Aim:

To perform Pandas Series

Description:

1. Create Series from array
2. Create series from dictionary
3. Create Series using scalar value
4. Create Series using index
5. Know the size, dimension, shape and index of Series

Program:

```
import pandas as pd
import numpy as np
#Creating Series from Array
arr = np.array(['P','a','n','d','a','s'])
a=pd.Series(arr)
print("Series from array:")
print(a)
```

Series from dictionary

```
nrr = {'x' : 0., 'y' : 1., 'z' : 2.}
b = pd.Series(nrr)
print("\n\nSeries from dictionary\n")
print(b)
```

Series using scalar

```
x = pd.Series(4, index=[0, 1, 2, 3])
print("\nSeries using scalar\n")
print(x)
```

```
#Series through index
x = pd.Series([1,2,3],index = ['a','b','c'])
print("\nSeries through index:")
print (x)
```

```
a=pd.Series(data=[1,2,3,4]) print("\n
a series:\n",a) print("\nIndex:
\n",a.index) print("\nValues:
\n",a.values) print("\nShape:
",a.shape) print("\nDimension:
",a.ndim)print("\nSize: ",a.size)
```

Output:

Series from array:

```
1  a
2  n
3  d
4  a
5  s
dtype: object
```

Series from dictionary:

```
y  1.0
z  2.0
dtype: float64
```

Series using scalar

```
0  4
```

1 4

2 4

3 4

dtype: int64

Series through index:

a 1

b 2

dtype: int64

n series:

0 1

1 2

2 3

3 4

dtype: int64Index:

RangeIndex(start=0, stop=4, step=1)

Values:

{1 2 3 4}

Shape: (4,)

Dimension: 1

Size: 4