



INDIAN SCHOOL AL WADI AL KABIR

Worksheet, 2020-21

Class: XII	SUB: INFORMATICS PRACTICES	Date of Completion:
Worksheet No:6	TOPIC : Pandas Dataframes Questions with Answers	15.05.2020

Q.I What are DataFrames?

Series			Series			DataFrame		
	apples			oranges		apples	oranges	
0	3		0	0	=	0	3	0
1	2	+	1	3		1	2	3
2	0		2	7		2	0	7
3	1		3	2		3	1	2

One limitation of series is that it is not able to handle data in the form of 2D or multidimensional related to real time.

For such tasks ,Python Pandas provides another data structure called DATAFRAMES

Ans: Dataframe objects of Pandas can store 2D heterogeneous data. It is a two dimensional data structure ,just like any table (with rows and columns).Dataframes are similar to spreadsheets or SQL tables .While working with Pandas ,dataframes are the most commonly used data structures.

Q.II Name the basic features of dataframes.

Ans:

- 1.Columns can be of different types.
- 2.Size of data frame is mutable,i.e, number of rows and columns can be increased or decreased any time
- 3.Its data/values are also mutable
- 4.Labelled axis(rows/columns)
- 5.Arithmetic operations on rows and columns
- 6.Indexes may constitute numbers ,strings and letters .

Q.III Name the constructs which can be used to create DataFrames.

Ans:

- Lists
- Series
- Dictionaries
- Numpy ndarrays

Q.IV Explain the Syntax of creating Dataframes.

Ans:

`pandas.DataFrame(data,index,columns,dtype,copySeries)`

- ✓ data: Data can be represented as series, list, dict, constants or other dataframes.
- ✓ index: For the row labels, the index to be used for the resulting frame is optional .By default ,the index value is displayed from 0 to n-1, if no index is passed.

- ✓ columns: For column labels, the optional default syntax is: np.arange(n). This is only true if no index is passed.
- ✓ dtype: Dtype is for data type of each column. If no data type is defined, None is applied.
- ✓ copy: This command is used for copying data if the default value is false.

Q.V. Give an example to show the creating dataframe from Lists.

Ans:

```
df111.py - C:/Users/Teacher/AppData/Local/Programs/Python/Python37-32/df111.py (3.7.4)
File Edit Format Run Options Window Help
import pandas as pd
list1=[5,10,15,20,25,30]
df=pd.DataFrame(list1)
print(df)
```

	0
0	5
1	10
2	15
3	20
4	25
5	30

#Naming the columns a dataframe.

```
import pandas as pd1
data1=[['Shiva',16],['Abdul',15],['David',17]]
df1=pd1.DataFrame(data1,columns=['Name','Age'])
print(df1)
```

	Name	Age
0	Shiva	16
1	Abdul	15
2	David	17

Q.VI. Show how to create DataFrame from Series.

```
import pandas as pd
s_Mark=pd.Series({'Vidya':80, 'Rahul':92, 'Megna':67, 'Radhika':95, 'Shaurya':97})
s_Age= pd.Series({'Vidya':32, 'Rahul':28, 'Megna':30, 'Radhika':25, 'Shaurya':20})
s_df=pd.DataFrame({'Marks':s_Mark, 'Age':s_Age})
```

The list of dictionaries can be passed as input data to create a data frame.
The dictionary keys, are by default, taken as column names.

	Marks	Age
Vidya	80	32
Rahul	92	28
Megna	67	30
Radhika	95	25
Shaurya	97	20

```
import pandas as pd
s={'Rinku':67, 'Ritu':78, 'Ajay':75, 'Pankaj':88, 'Aditya':92},
  {'Rinku':77, 'Ritu':58, 'Ajay':87, 'Pankaj':65},
  {'Rinku':88, 'Ajay':67, 'Pankaj':74, 'Aditya':70}}
ndf=pd.DataFrame(s)
print(ndf)
```

	Rinku	Ritu	Ajay	Pankaj	Aditya
0	67	78.0	75	88	92.0
1	77	58.0	87	65	NaN
2	88	NaN	67	74	70.0

Q.VII-a Show with an example how data can be sorted in the dataframe.

```
import pandas as pd
s_Mark=pd.Series({'Rinku':67,'Ritu':78,'Ajay':75,'Pankaj':88,'Aditya':92})
s_Age=pd.Series({'Rinku':15,'Ritu':17,'Ajay':16,'Pankaj':15,'Aditya':18})
ndf=pd.DataFrame({'Marks':s_Mark,'Age':s_Age})
print(ndf.sort_values(by=['Marks']))
```

	Marks	Age
Rinku	67	15
Ajay	75	16
Ritu	78	17
Pankaj	88	15
Aditya	92	18

#By default sorting is done by ascending order.

Following is an example of sorting data in descending order.

```
import pandas as pd
s_Mark=pd.Series({'Rinku':67,'Ritu':78,'Ajay':75,'Pankaj':88,'Aditya':92})
s_Age=pd.Series({'Rinku':15,'Ritu':17,'Ajay':16,'Pankaj':15,'Aditya':18})
ndf=pd.DataFrame({'Marks':s_Mark,'Age':s_Age})
print(ndf.sort_values(by=['Marks'],ascending=False))
```

	Marks	Age
Aditya	92	18
Pankaj	88	15
Ritu	78	17
Ajay	75	16
Rinku	67	15

Q.VII -b How to create a DataFrame by passing list of dictionaries.

```
import pandas as pd
s={'Name':['Rinku','Ritu','Ajay','Pankaj','Aditya'],
  'English':[67,78,75,88,92],
  'Economics':[78,67,89,90,56],
  'Info':[78,88,98,90,87],
  'Accounts':[77,70,80,67,86]}
print("The series is")
```

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

#output of print(s)

```
The series is
{'Name': ['Rinku', 'Ritu', 'Ajay', 'Pankaj', 'Aditya'], 'English': [67, 78, 75, 88, 92], 'Economics': [78, 67, 89, 90, 56], 'Info': [78, 88, 98, 90, 87], 'Accounts': [77, 70, 80, 67, 86]}
```

#output of print(df)

	Name	English	Economics	Info	Accounts
0	Rinku	67	78	78	77
1	Ritu	78	67	88	70
2	Ajay	75	89	98	80
3	Pankaj	88	90	90	67
4	Aditya	92	56	87	86

Q.VIII How to create an indexed dataframe?

Ans:

```
import pandas as pd
```

```
s={
    'Name':['Rinku','Ritu','Ajay','Pankaj','Aditya'],
    'English':[67,78,75,88,92],
    'Economics':[78,67,89,90,56],
    'Info':[78,88,98,90,87],
    'Accounts':[77,70,80,67,86]
}
```

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

```
print(df)
```

	Name	English	Economics	Info	Accounts
0	Rinku	67	78	78	77
1	Ritu	78	67	88	70
2	Ajay	75	89	98	80
3	Pankaj	88	90	90	67
4	Aditya	92	56	87	86

#part-II

```
import pandas as pd
```

```
s={
    'Name':['Rinku','Ritu','Ajay','Pankaj','Aditya'],
    'English':[67,78,75,88,92],
    'Economics':[78,67,89,90,56],
    'Info':[78,88,98,90,87],
    'Accounts':[77,70,80,67,86] }
df=pd.DataFrame(s,index=['Sno1', 'Sno2','Sno3','Sno4','Sno5'])
```

```
print(df)
```

	Name	English	Economics	Info	Accounts
Sno1	Rinku	67	78	78	77
Sno2	Ritu	78	67	88	70
Sno3	Ajay	75	89	98	80
Sno4	Pankaj	88	90	90	67
Sno5	Aditya	92	56	87	86

Q.IX How to change the index or row names?

Ans:

Create an example dataframe

import pandas as pd

data = {'Commander': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],

 'Date': ['2012, 02, 08', '2012, 02, 08', '2012, 02, 08', '2012, 02, 08', '2012, 02, 08'],

 'Score': [4, 24, 31, 2, 3]}

df = pd.DataFrame(data, index = ['Cochice', 'Pima', 'Santa Cruz', 'Maricopa', 'Yuma'])

print(df)

	Commander	Date	Score
Cochice	Jason	2012, 02, 08	4
Pima	Molly	2012, 02, 08	24
Santa Cruz	Tina	2012, 02, 08	31
Maricopa	Jake	2012, 02, 08	2
Yuma	Amy	2012, 02, 08	3

Q.X Program to change the Index with a column name.

Ans:

import pandas as pd

s={'Name':['Rinku','Ritu','Ajay','Pankaj','Aditya'],

 'English':[67,78,75,88,92],

 'Economics':[78,67,89,90,56],

 'Info':[78,88,98,90,87],

 'Accounts':[77,70,80,67,86]}

df=pd.DataFrame(s)

df.set_index('Name',inplace=True)

print(df)

Name	English	Economics	Info	Accounts
Rinku	67	78	78	77
Ritu	78	67	88	70
Ajay	75	89	98	80
Pankaj	88	90	90	67
Aditya	92	56	87	86

Q.XI Illustrate with an example program ,how to rename a column name in a DataFrame?

Ans:

import pandas as pd

a1=[20,30,25,26,15]

```
#converting age list into dataframe
df=pd.DataFrame(a1)
print(df)
```

```
0
0 20
1 30
2 25
3 26
4 15
```

```
#part-II
import pandas as pd
a1=[20,30,25,26,15]
#converting age list into dataframe
df=pd.DataFrame(a1)
df.columns=['Age']
df['Age2']=45
print(df)
```

```
Age Age2
0 20 45
1 30 45
2 25 45
3 26 45
4 15 45
>>>
```

Q.XII How to add a new column to a dataframe?

```
import pandas as pd
a1=[20,30,25,26,15]
#converting age list into dataframe
df=pd.DataFrame(a1)
df.columns=['Age']
df['Age2']=45
df['Age3']=pd.Series([42,44,50,60,45],index=[0,1,2,3,4])
df['Total']=df['Age']+df['Age2']+df['Age3']
print(df)
```

```
Age Age2 Age3 Total
0 20 45 42 107
1 30 45 44 119
2 25 45 50 120
3 26 45 60 131
4 15 45 45 105
```

Q.XIII How to update the column values in a dataframe?

Ans:

```
import pandas as pd
a1=[20,30,25,26,15]
#converting age list into dataframe
df=pd.DataFrame(a1)
df.columns=['Age']
df['Age2']=45
df['Age3']=pd.Series([42,44,50,60,45],index=[0,1,2,3,4])
df['Total']=df['Age']+df['Age2']+df['Age3']
df['Total']=df['Total']+10
df['updated _Age']=df['Total']
print(df)
```

```
Age Age2 Age3 Total updated _Age
0 20 45 42 117 117
1 30 45 44 129 129
2 25 45 50 130 130
3 26 45 60 141 141
4 15 45 45 115 115
```

Q.XIV Program for selecting a column from DataFrame.

Ans:

```
import pandas as pd
a1=[20,30,25,26,15]
#converting age list into dataframe
df=pd.DataFrame(a1)
df.columns=['Age']
df['Age2']=45
df['Age3']=pd.Series([42,44,50,60,45],index=[0,1,2,3,4])
df['Total']=df['Age']+df['Age2']+df['Age3']
df['Total']=df['Total']+10
df['updated _Age']=df['Total']
print(df.iloc[:,[0,3]])
```

	Age	Total
0	20	117
1	30	129
2	25	130
3	26	141
4	15	115

Q.XV Add this statement to above (Q.XIV) and guess the output:

```
print(df.iloc[:,0:4])
```

Output:

	Age	Age2	Age3	Total	updated _Age
0	20	45	42	117	117
1	30	45	44	129	129
2	25	45	50	130	130
3	26	45	60	141	141
4	15	45	45	115	115

Q.XVI What are the different ways of deleting a column from DataFrame.

```
Ans: import pandas as pd
a1=[20,30,25,26,15]
#converting age list into dataframe
df=pd.DataFrame(a1)
df.columns=['Age']
df['Age2']=45
df['Age3']=pd.Series([42,44,50,60,45],index=[0,1,2,3,4])
df['Total']=df['Age']+df['Age2']+df['Age3']
df['Total']=df['Total']+10
df['updated _Age']=df['Total']
del df['updated _Age']
print(df)
```

	Age	Age2	Age3	Total
0	20	45	42	117
1	30	45	44	129
2	25	45	50	130
3	26	45	60	141
4	15	45	45	115

Q.XVII Add this statement to the above coding and guess the output.

```
df=df.pop('Age3')
print(df)
```

Output:

```
0  42
1  44
2  50
3  60
4  45
Name: Age3, dtype: int64
```

Q.XVIII Guess the output after adding the following statement in the coding of Q.XVI

```
df=df.drop('Age2',axis=1)
print(df)
```

Output:

```
   Age  Age3  Total  updated_Age
0   20   42   117         117
1   30   44   129         129
2   25   50   130         130
3   26   60   141         141
4   15   45   115         115
```

Q.XIX Guess the output after adding the following statement to the coding of Q.XVI

```
df=df.drop(0,axis=0)
```

Q.XX What is binary operations ?illustrate with an example on addition,subtraction, multiplication and division operation.

Ans:

#Binary operations on DataFrame

```
import pandas as pd
```

```
studentA={'Unit Test-1':[5,6,8,3,10],'Unit Test-2':[7,8,9,6,10]}
```

```
studentB={'Unit Test-1':[3,3,6,6,8],'Unit Test-2':[5,9,8,10,5]}
```

```
ds1=pd.DataFrame(studentA)
```

```
ds2=pd.DataFrame(studentB)
```

```
print(ds1)
```

```
print(ds2)
```

```
print("subtraction")
```

```
print(ds1.sub(ds2))
```

```
print("rsub")
```

```
print(ds1.rsub(ds2))
```

```
   Unit Test-1  Unit Test-2
0           5           7
1           6           8
2           8           9
3           3           6
4          10          10
```

```
   Unit Test-1  Unit Test-2
0           3           5
1           3           9
2           6           8
3           6          10
4           8           5
```


subtraction		
	Unit Test-1	Unit Test-2
0	2	2
1	3	-1
2	2	1
3	-3	-4
4	2	5

rsub		
	Unit Test-1	Unit Test-2
0	-2	-2
1	-3	1
2	-2	-1
3	3	4
4	-2	-5

Q. XXI Use of add() and radd() operation.

Ans:

#Binary operations on DataFrame

import pandas as pd

studentA={'Unit Test-1':[5,6,8,3,10],'Unit Test-2':[7,8,9,6,10]}

studentB={'Unit Test-1':[3,3,6,6,8],'Unit Test-2':[5,9,8,10,5]}

ds1=pd.DataFrame(studentA)

ds2=pd.DataFrame(studentB)

print(ds1)

print(ds2)

print("Addition")

print(ds1.add(ds2))

print("radd")

print(ds1.radd(ds2))

Addition		
	Unit Test-1	Unit Test-2
0	8	12
1	9	17
2	14	17
3	9	16
4	18	15

radd		
	Unit Test-1	Unit Test-2
0	8	12
1	9	17
2	14	17
3	9	16
4	18	15

Q.XXII Using Mul() and div() .

Ans:

#Binary operations on DataFrame

import pandas as pd

studentA={'Unit Test-1':[5,6,8,3,10],'Unit Test-2':[7,8,9,6,10]}

studentB={'Unit Test-1':[3,3,6,6,8],'Unit Test-2':[5,9,8,10,5]}

ds1=pd.DataFrame(studentA)

ds2=pd.DataFrame(studentB)

print(ds1)

print(ds2)

print("Multiplication")

print(ds1.mul(ds2))

print("Division")

print(ds1.div(ds2))

	Unit Test-1	Unit Test-2
0	5	7
1	6	8
2	8	9
3	3	6
4	10	10
	Unit Test-1	Unit Test-2
0	3	5
1	3	9
2	6	8
3	6	10
4	8	5

Multiplication		
	Unit Test-1	Unit Test-2
0	15	35
1	18	72
2	48	72
3	18	60
4	80	50

Division		
	Unit Test-1	Unit Test-2
0	1.666667	1.400000
1	2.000000	0.888889
2	1.333333	1.125000
3	0.500000	0.600000
4	1.250000	2.000000

Q.XXIII Sample questions for -MCQ online test.

1. What do you mean by `df.iloc[:,[0,3]]` ?

- a. To access 0th column and 3rd columns from the dataframe df.
- b. It is an invalid command.
- c. To access first and fourth column from the dataframe df.
- d. To access all the columns between first and fourth .

Answer: c

2. Identify the 3 ways of deleting column/s from the existing dataframe df.

- a. `delete_it df[]` `drop[labels,axis=1]` `df.pop(label)`
- b. `del df[labels]` `drop(labels , axis=2)` `df.pop(label)`
- c. Both a. and b. are correct
- d. `del df[labels]` `drop(labels , axis=0)` `df.pop(label)`

Answer: d

3. What is `df["Total"]=pd.Series([1000,2000,3000,4000],index=[0,1,2,3])` ?

- a. New row is added named Total with given values to the dataframe df.
- b. New column named Total is added to dataframe df with the given values.
- c. It is renaming a column with Total and assigning values.
- d. Both a. and b. are correct.

Answer: b