Advanced Operations on DataFrames



Informatics Practices, Class-12



CHAPTER-2

By: Mrs. Neha Tyagi (PGT CS) KV No-5 2nd Shift, Jaipur KVS RO Jaipur

Pivoting DataFrame

- Pandas is a popular library for *Data analysis*.
- *Pivoting* is one of the key actions for a Data–Analyst. Means providing an axis to the table data, on the basis of that axis the database will work.
- Using Pandas, MS-Excel type of pivot tables can be created.
- These tables *summarizes* the big data and create meaningful reports to save your time.
- Pivot table allows us to fetch important record from a large and detailed data set.
- Pivot tables can automatically sort, count and total etc.
- In general, pivoting means to use unique value from a index/column and make dataframe.
- To make pivot table we use *pivot()* or *pivot_table()* from pandas.

Pivoting using pivot() method

- pivot() method, creates new DataFrame after reshaping the data on the basis of column values.
- This method takes 3 arguments *index*, *columns* and *values*. Minimum two arguments are compulsory.
- In the form of arguments value you have to pass column name of original table.
- Then pivot () creates a new table whose indices of row and column are the same which you have given as argument.
- Cell values of new table will come from the column which you have given as parameter. Its syntax is -

pandas.pivot(index, columns, values)

- Where *index* creates a index of new DataFrame, which is the column name from the table.
- Where *columns* creates columns of new DataFrame, which are the names of column of table.
- Where *values* creates columns of new DataFrame which are the values of the column name from table.

Pivoting using pivot() method

syntax → pandas.pivot(index, columns, values)

• Example → Creating DataFrame

>> df Name Subject Score Grade Pratibha CS 99 A1 Ritika Phv 87 A2 Saumya Chem 88 A2 67 В Aryan Maths



• Creating pivot table \rightarrow

>>> pv=df	<pre>.pivot</pre>	(index	='Name'	,columr	ns='Subject',values='Score')
>>> pv					
Subject	CS	Chem	Maths	Phy	
Name				-	We can see in this pivot table that there
Aryan	NaN	NaN	67.0	NaN	is a new table is created and the values
Pratibha	99.0	NaN	NaN	NaN	of Score column came in to different
Ritika	NaN	NaN	NaN	87.0	columns. While its Name and Subject
Saumya	NaN	88.0	NaN	NaN	column, is matching with original table

Where values are not matching, NaN (None) is putted automatically.

Using pivot() method with .fillna()

syntax → pandas.pivot(index, columns, values).fillna()

• Example → Creating DataFrame.

Score Grade Name Subject Pratibha CS 99 A1 Ritika Phv 87 A2 Saumya Chem 88 A2 В Aryan Maths 67



Creating pivot table with .fillna()→

>>> pv=df	.pivot(index=')	Name',columns='	<pre>Subject',values='Score').fillna('')</pre>
>>> pv			
Subject	CS Chem Maths	Phy	
Name		-	We can see in this pivot table that there
Aryan	67		is a new table is created and the values
ni yan Duutibbu	00		of Score column came in to different
Pratibna	99		columns. While its Name and Subject
Ritika		87	column, is matching with original table.
Saumya	88		Where values are not matching, there
			is a blank space in spite of NaN.

	Pivo	oting	by N	Multij	ole d	colu	mns		
We remo	ve <i>va</i>	lues	param	neter f	rom s	synta	ix only	/.	
syntax	$x \rightarrow pa$	andas.	pivot(ir	ndex, c	olumn	s)	>>> df		
 Example 	Pivoting by Multiple columns Veremove values parameter from syntax only. syntax -> pandas.pivot(index, columns) Example -> Creating DataFrame mport pandas as pd 'lassXII={'Name':['Pratibha', 'Ritika', 'Saumya', 'Aryan'], \ 'Subject':['CS', 'Phy', 'Chem', 'Maths'], \ 'Score':[99,87,88,67], \ 'Grade':['A1', 'A2', 'A2', 'B']} if=pd.DataFrame(ClassXII, columns=['Name', 'Subject', 'Score', 'Grade'] Ceating pivot table with .fillna() -> >>> pv=df.pivot(index='Name', columns='Subject') >>> pv Score Grade Subject CS Chem Maths Aryan NaN Aryan NaN	Score Grade 99 A1 87 A2							
import pan	das <mark>as</mark> j	pd					2 Sau 3 Ar	umya Chem ryan Maths	88 A2 67 B
df=pd.Data • Ceating	'Subjec 'Score' 'Grade' Frame (C pivot ta	:[99,8 :['A1', lassXII	7,88,67 ,'A2',' I,colum th.fillna	','Chem],\ A2','B' ns=['Na a()→	','Mat.]} me','S	ubject	:','Scor	ce','Gi	rade'])
>>> pv=df	E.pivot	(index	x='Name	',colu	mns='s	Subjed	ct')		
>>> pv	_								
	Score				Grade				
Subject	CS	Chem	Maths	Phy	CS	Chem	Maths	Phy	
Name									
Aryan	NaN	NaN	67.0	NaN	NaN	NaN	В	NaN	
Pratibha	99.0	NaN	NaN	NaN	A1	NaN	NaN	NaN	
Ritika	NaN	NaN	NaN	87.0	NaN	NaN	NaN	A2	
Saumya	NaN	88.0	NaN	NaN	NaN	A2	NaN	NaN	

Pivoting by Multiple columns. . .

- In last example we have seen that there are many indices created and their values were seen once for subjects and once for grades for each name.
- We can filter them \rightarrow

>>> pv.Sc	ore.fillna('	')	<pre>>>> pv.Score.CS.fillna('')</pre>
Subject	CS Chem Mat	hs Phy	Name
Name			Aryan
Aryan		67	Pratibha 99
Pratibha	99		Ritika
Ritika		87	Saumya
Saumya	88		Name: CS, dtype: object
>>> pv.Gr	ade.CS.fill	na('')	>>> pv.Grade.Maths.fillna('')
Name			Name
Aryan			Aryan B
Pratibha	A1		Pratibha
Ritika			Ritika
Saumya			Saumya
Name: CS,	dtype: obj	ect	Name: Maths, dtype: object

Neha Tyagi, KV No-5 Jaipur

Pivot Problem

• We should always remember that if there are combinations of multiple values in indices and columns then a value error will occur.

```
import pandas as pd
ClassXII={'Name':['Pratibha','Ritika','Saumya','Aryan','Pratibha'],\
            'Subject': ['CS', 'Phy', 'Chem', 'Maths', 'CS'], \
            'Score': [99,87,88,67,98],
            'Grade':['A1', 'A2', 'A2', 'B', 'A1']}
df=pd.DataFrame(ClassXII,columns=['Name','Subject','Score','Grade'])
>>> df
      Name Subject Score Grade
  Pratibha
0
               CS
                      99
                           A1
1
    Ritika Phy 87
                        A2
2
    Saumya Chem 88 A2
3
     Aryan Maths 67
                          В
  Pratibha
                      98
4
               CS
                           Α1
>>> pv=df.pivot(index='Name',columns='Subject',values='Score')
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
   pv=df.pivot(index='Name', columns='Subject', values='Score')
  File "C:\Users\KVBBKServer\AppData\Local\Programs\Python\Python36\lib\site-packag
es\pandas\core\frame.py", line 5194, in pivot
    return pivot(self, index=index, columns=columns, values=values)
  File "C:\Users\KVBBKServer\AppData\Local\Programs\Python\Python36\lib\site-packag
es\pandas\core\reshape\reshape.py", line 415, in pivot
      turn indexed unstack (solumns)
```

ValueError: Index contains duplicate entries, cannot reshape

Using stack() and unstack() methods

 stack() and unstack() methods both flip the layout of DataFrame, means these flips the levels of columns into row and flips levels of rows into columns. DataFrame*stacking* means moving the innermost column index to innermost row index and the opposite action is know as *unstacking*

```
import pandas as pd
ClassXII={'Name':['Pratibha','Ritika','Saumya','Aryan'],\
             'Subject': ['CS', 'Phy', 'Chem', 'Maths'], \
             'Score': [99,87,88,67], \
             'Grade': ['A1', 'A2', 'A2', 'B'] }
df=pd.DataFrame(ClassXII,columns=['Name','Subject','Score','Grade'])
>>> pv=df.pivot(index='Name',columns='Subject')
                                                     Using Stack () Method
>>> pv
        Score
                             Grade
                                                   >>> pv.stack()
              Chem Maths
                                CS Chem Maths
Subject
          CS
                          Phy
                                              Phy
                                                                       Score Grade
Name
                                                             Subject
Aryan
         NaN
                    67.0
                          NaN
                                    NaN
                                             NaN
                                                   Name
               NaN
                               NaN
                                           В
Pratibha
         99.0
                    NaN
               NaN
                          NaN
                                A1
                                    NaN
                                         NaN
                                             NaN
                                                             Maths
                                                                        67.0
                                                   Aryan
                                                                                  В
Ritika
         NaN
                    NaN
                         87.0
                                         NaN
                                              A2
               NaN
                               NaN
                                    NaN
                                                   Pratibha
                                                             CS
                                                                        99.0
                                                                                 Α1
Saumya
         NaN
              88.0
                    NaN
                          NaN
                               NaN
                                     A2
                                         NaN
                                             NaN
                                                   Ritika
                                                             Phy
                                                                        87.0
                                                                                 A2
After using Stack Method all the horrizontal
                                                   Saumya
                                                             Chem
                                                                        88.0
                                                                                 A2
became vertical and it takes last level in column
breakdown and converts it into last row breakdown.
```

Using stack() and unstack() methods...

import	pand	las <mark>a</mark>	s pd											
ClassXI	[]={ '	Name	':[']	Prati	bha '	,'Ri	itika	','	Saumya','	Aryan']	, \			
'Subject':['CS','Phy','Chem','Maths'],\														
	'Score':[99,87,88,67],\													
'Grade':['A1','A2','A2','B']}														
f=pd.DataFrame(ClassXII,columns=['Name','Subject','Score','Grade'])														
<pre>>> pv=df.pivot(index='Name',columns='Subject') >> pv</pre>														
Score Grade Score Grade														
Subject	CS	Chem	Maths	Phy	CS	Chem	Maths	Phy	Name	Subject				
Jame							-		Aryan	Maths	67.0	В		
Aryan Pratibba	NaN 99 A	NaN NaN	6/.U NaN	NaN NaN	NaN 1	NaN NaN	N a N	NaN NaN	Pratibha	CS	99.0	A1		
Ritika	NaN	NaN	NaN	87.0	NaN	NaN	NaN	A2	Ritika	Phy	87.0	A2		
Saumya	NaN	88.0	NaN	NaN	NaN	A2	NaN	NaN	Saumya	Chem	88.0	A2		
>>> pv.	stack	().st	ack()	_		Stac	rk can	hei	ised like th	is After s	tacking t	here is		
Name	Su	bject				onat	hor of		age than it		l the rem			
Aryan	Ma	ths	Sco	ore	67		iner si	lacki	ng, men it	moves an	i the ren	aning		
			Gra	ade	В	leve	IS.							
Pratibh	a CS	5	Sco	ore	99									
			Gra	ade	A1									
Ritika Phy Score 87														
			Gra	ade	A2									
Saumya	Ch	em	Sco	ore	88									
			Gra	ade	A2									
dtype:	objec	:t												

Using stack() and unstack() methods...

ort	pano	las as	s pa											
ISSXI	I={	'Name'	':[']	Prati	bha '	, 'R	itika	· , ' s	Saumya',	'Arya	in']	, \		
		'Subie	ect'	:['CS	'.'F	hv'	.'Che	em'.	'Maths'l	.\				
		'Score	- · · [(99 87	88	671	\]	, ,				
			- • L -	JJ,07	,00,		/ \ 2 D	111						
1 -		Grade		AL,	AZ	, A.	Z', 'B)] }						
=pd.L	atal	rame	(CLa	SSXII	, col	umn	s=['N	lame	','Subje	ct','	Sco	re',	'Grad	e'])
pv=df	pivot	.(index=	-'Name	'.colur	mns='S	Subied	ct')		>>> pv.st	cack(0)				
pv		(1110011		,			,		Subject		CS	Chem	Maths	Phy
	Score			(Grade				Name					
ect	CS	Chem M	laths	Phy	CS	Chem	Maths	Phy	Aryan	Grade	NaN	NaN	В	NaN
2	NT - NT	NT - NT	67 0	NT - NT	NT - NT	NT - NT	D	NT - NT		Score	NaN	NaN	67	NaN
in Tibba		NaN NaN	67.U NoN	NaN NaN	NaN 1	NaN NaN	N D N	NaN NaN	Pratibha	Grade	A1	NaN	NaN	NaN
ka	NaN	NaN	NaN	87.0	NaN	NaN	NaN	A2	Ditile	Score	99 NoN	NaN	NaN	NaN
iya	NaN	88.0	NaN	NaN	NaN	A2	NaN	NaN	RILIKA	Grade	NoN	NaN	NaN NaN	AZ Q7
-	- at a	alt(0)	ata	~lr ()					Saumva	Grade	NaN	A2	NaN	NaN
>> b/	/.Sta	ICK(0)	.stac	ск()					Saamya	Score	Nail	88	NaN	NaN
ame			Sub	oject					I					
ryan		Grade	Mat	ths		В					hie ie	une	ackino	
		Score	Mat	ths	6	57				_				
ratik	bha	Grade	CS		P	1								
		Score	CS		9	9	Uns	stac	<i>kina</i> is i	ust like	e stad	ck the)	
itika	ì	Grade	Phy	7	P	2		diffo		that th				
		Score	Phy	J	6	37	Only	une			iere			
aumva	4	Grade	Che	ر m<	7	2	argur	nent	'0' is pa	ssed i	n sta	ack()	
aamyc	•	Score	Che	em	5	88	meth	od.						
turno	obi	Act			_	-								
	<pre>pd.E pv=df pv=df pv ect ibha ka ya >> pv ame ryan ratik itika aumya</pre>	<pre>pd.Datal pv=df.pivot pv Score ect CS n NaN ibha 99.0 ka NaN ibha 90.0 ka NaN ibha 90.</pre>	<pre>issXII={'Name' 'Subje 'Score 'Grade pv=df.pivot(index= pv Score ect CS Chem N ibha 99.0 NaN ibha 99.0 NaN ibha 99.0 NaN ka NaN NaN iya NaN 88.0 >> pv.stack(0) ame ryan Grade Score itika Grade Score itika Grade Score aumya Grade Score </pre>	<pre>issXII={'Name': ['] 'Subject': 'Score': [9 'Grade': [pv=df.pivot(index='Name pv Score ect CS Chem Maths n NaN NaN 67.0 ibha 99.0 NaN NaN ka NaN NaN NaN ka NaN NaN NaN ka NaN NaN NaN ka NaN NaN NaN iya NaN 88.0 NaN >> pv.stack(0).stac ame Suk ryan Grade Mat Score Mat ratibha Grade CS Score CS itika Grade Phy aumya Grade Che Score Che </pre>	<pre>issXII={'Name':['Prati 'Subject':['CS 'Score':[99,87 'Grade':['A1', pd.DataFrame(ClassXII pv=df.pivot(index='Name',colum pv Score ect CS Chem Maths Phy in NaN NaN 67.0 NaN ibha 99.0 NaN NaN ST.0 ibha 99.0 NaN NaN NaN ibha 99.0 Score CS itika Grade CS itika Grade Phy score CS itika Grade Chem score Chem tupot object</pre>	<pre>Solution pandas as pd issXII={'Name': ['Pratibha' 'Subject': ['CS', 'F 'Score': [99,87,88, 'Grade': ['A1', 'A2' pd.DataFrame(ClassXII,col pv=df.pivot(index='Name',columns='S pv Score Grade ect CS Chem Maths Phy CS n NaN NaN 67.0 NaN NaN ibha 99.0 NaN NaN NaN A1 ka NaN NaN 67.0 NaN NaN ibha 99.0 NaN NaN NaN A1 ka NaN NaN NaN 87.0 NaN ya NaN 88.0 NaN NaN NaN >> pv.stack(0).stack() ame Subject ryan Grade Maths Score CS 9 itika Grade CS 7 Score CS 9 itika Grade Phy 7 Score Phy 8 aumya Grade Chem 7 Score Chem 8</pre>	<pre>Solution pandas as pu ssXII={'Name': ['Pratibha', 'R. 'Subject': ['CS', 'Phy' 'Score': [99,87,88,67] 'Grade': ['A1', 'A2', 'A. pd.DataFrame (ClassXII, columns pv=df.pivot(index='Name', columns='Subject pv score Grade ect CS Chem Maths Phy CS Chem .n NaN NaN 67.0 NaN NaN NaN ibha 99.0 NaN NaN NaN A1 NaN ka NaN NaN NaN 87.0 NaN NaN ka NaN NaN NaN 87.0 NaN NaN ya NaN 88.0 NaN NaN NaN A2 >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score Maths 67 ratibha Grade CS A1 Score CS 99 itika Grade Phy A2 Score Phy 87 aumya Grade Chem A2 Score Chem 88 tupo: object</pre>	<pre>IssXII={'Name':['Pratibha','Ritika 'Subject':['CS','Phy','Che 'Score':[99,87,88,67],\ 'Grade':['A1','A2','A2','E pd.DataFrame(ClassXII,columns=['N pv=df.pivot(index='Name',columns='Subject') pv Score Grade ect CS Chem Maths Phy CS Chem Maths in NaN NaN 67.0 NaN NaN NaN B ibha 99.0 NaN NaN NaN A1 NaN NaN ka NaN NaN 67.0 NaN NaN NaN NaN ka NaN NaN 87.0 NaN NaN NaN ka NaN NaN NaN 87.0 NaN NaN NaN ka NaN NaN NaN 87.0 NaN NaN NaN pya NaN 88.0 NaN NaN NaN A2 NaN >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score CS 99 itika Grade CS A1 Score CS 99 itika Grade Phy A2 Score Phy 87 aumya Grade Chem A2 Score Chem 88</pre>	<pre>Solution painda's as put issXII={'Name': ['Pratibha', 'Ritika', 'S 'Subject': ['CS', 'Phy', 'Chem', ' 'Score': [99,87,88,67], 'Grade': ['A1', 'A2', 'A2', 'B']} pd.DataFrame(ClassXII, columns=['Name' pv=df.pivot(index='Name', columns='Subject') pv Score Grade ect CS Chem Maths Phy CS Chem Maths Phy in NaN NaN 67.0 NaN NaN NaN B NaN ibha 99.0 NaN NaN NaN A1 NaN NaN NaN ka NaN NaN 67.0 NaN NaN NaN NaN NaN ka NaN NaN 67.0 NaN NaN NaN NaN A2 ya NaN 88.0 NaN NaN NaN NaN A2 NaN 88.0 NaN NaN NaN A2 NaN NaN >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score CS 99 itika Grade CS A1 Score CS 99 itika Grade Phy A2 Score Chem 88 tuma: object tuma: object</pre>	<pre>Solution pandas as put ssXII={'Name': ['Pratibha', 'Ritika', 'Saumya',</pre>	<pre>Solution pandas as pd ssXII={'Name':['Pratibha','Ritika','Saumya','Arya 'Subject':['CS','Phy','Chem','Maths'],\ 'Score':[99,87,88,67],\ 'Grade':['A1','A2','A2','B']} pd.DataFrame(ClassXII,columns=['Name','Subject',' pv=df.pivot(index='Name',columns='Subject') pv score Grade ect CS Chem Maths Phy CS Chem Maths Phy score Grade ect CS Chem Maths Phy CS Chem Maths Phy in NaN NaN 67.0 NaN NaN NaN B NaN ibha 99.0 NaN NaN NaN A1 NaN NaN NaN ka NaN NaN 67.0 NaN NaN NaN NaN A2 NaN 88.0 NaN NaN A1 NaN NaN NaN ka NaN NaN 87.0 NaN NaN NaN A2 NaN 88.0 NaN NaN NaN A2 NaN NaN >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score CS 99 itika Grade CS A1 Score CS 99 itika Grade Phy A2 Score Phy 87 aumya Grade Chem A2 Score Chem 88 turet shiect</pre>	<pre>Nort pandas as put ssXII={'Name':['Pratibha','Ritika','Saumya','Aryan'], 'Subject':['CS','Phy','Chem','Maths'], 'Score':[99,87,88,67], 'Grade':['A1','A2','A2','B']} pd.DataFrame(ClassXII,columns=['Name','Subject','Scon pv=df.pivot(index='Name',columns='Subject') pv score Grade ect CS Chem Maths Phy CS Chem Maths Phy in NaN NaN 67.0 NaN NAN NAN B NAN ibha 99.0 NAN NAN NAN A1 NAN NAN NAN ka NAN NAN 67.0 NAN NAN NAN B NAN ibha 99.0 NAN NAN NAN A1 NAN NAN NAN ka NAN NAN 88.0 NAN NAN A1 NAN NAN NAN ka NAN NAN 88.0 NAN NAN A2 NAN NAN A2 nya NAN 88.0 NAN NAN NAN A2 NAN NAN >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score CS 99 itika Grade CS A1 Score CS 99 itika Grade Phy A2 Score Phy 87 aumya Grade Chem A2 Score Chem 88 turet shiadt</pre>	<pre>Nore pandas as pd assXII={'Name': ['Pratibha', 'Ritika', 'Saumya', 'Aryan'], 'Subject': ['CS', 'Phy', 'Chem', 'Maths'], 'Score': [99,87,88,67], 'Grade': ['A1', 'A2', 'A2', 'B']} spd.DataFrame(ClassXII, columns=['Name', 'Subject', 'Score', pv=df.pivot(index='Name', columns='Subject') pv Score Grade ect CS Chem Maths Phy CS Chem Maths Phy in NaN NaN 67.0 NaN NaN NaN B NaN ibha 99.0 NaN NaN NaN A1 NaN NaN NaN ka NaN NaN NaN 67.0 NaN NaN NaN NaN NaN ya NaN 68.0 NaN NaN NaN A2 NaN NaN ya NaN 68.0 NaN NaN NaN A2 NaN NaN >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score CS 99 itika Grade CS A1 Score CS 99 itika Grade Phy A2 Score Phy 87 aumya Grade Chem A2 Score Chem 88 turpet object turpet ob</pre>	<pre>sole pandas as pa ssXII={'Name': ['Pratibha', 'Ritika', 'Saumya', 'Aryan'], 'Subject': ['CS', 'Phy', 'Chem', 'Maths'], 'Score': [99,87,88,67], 'Grade': ['A1', 'A2', 'A2', 'B']} pd.DataFrame(ClassXII,columns=['Name', 'Subject', 'Score', 'Grad pv=df.pivot(index='Name',columns='Subject') pv Score Grade ect CS Chem Maths Phy CS Chem Maths Phy in NaN NaN 67.0 NaN NaN NaN B NaN ibha 99.0 NaN NaN NaN A1 NaN NaN NaN ka NaN NaN 67.0 NaN NaN NaN NaN NaN ka NaN NaN 87.0 NaN NaN NaN NaN ka NaN NaN 88.0 NaN NaN NaN A2 NaN ya NaN 88.0 NaN NaN NaN A2 NaN NaN >> pv.stack(0).stack() ame Subject ryan Grade Maths B Score Maths 67 ratibha Grade CS A1 Score CS 99 itika Grade Phy A2 Score Phy 87 aumya Grade Chem A2 Score Chem 88 turpation for the stack for the stac</pre>

Neha Tyagi, KV No-5 Jaipur

- This is the generalization of pivot() method.
- When you have duplicate values for only one index or duplicate values for one column then pivot_table() method is used.
- A pivot table contains counts, sums and table data related functions.
- pivot_table() method creates a DataFrame, kind of Excel Sheet.
- This method is used to convert row into column and vice-versa.
- It allows grouping of any data field.
- Its syntax is →
- pandas.pivot_table (DataFrame, values=None, index=None,

columns=None, aggfunc='mean', fill_value=None, margins=False, dropna=True, margins_name='All')

• All the arguments are not necessary in .pivot_table() method, because there are some default values for some arguments.

pandas.pivot_table (DataFrame, values=None, index=None,

columns=None, aggfunc='mean', fill_value=None, margins=False, dropna=True, margins_name='All')

- All the arguments are not necessary in .pivot_table() method, because there are some default values for some arguments.
- In its syntax -
 - **DataFrame** \rightarrow is a pandas DataFrame.
 - values \rightarrow this is optional and also a column to be aggregated.
 - index \rightarrow this is column, grouper, array or list name.
 - columns \rightarrow this is a column, grouper, array or list.
 - **aggfunc** \rightarrow is an aggregation function.
 - fill_value→ we can set default values using this, if the values are not given.
 - margins
 this is a boolean whose default is false. If we make it true then the sum of row and column in resulting dataframe.
 - dropna \rightarrow if this is true then it drops row having missing data
 - margins_name='All' → if margins is true then it keeps the name of the rows and column of total.
 Neha Tyagi, KV No-5 Jaipur

We create a pivot table considering the following data. \rightarrow



>>> pv=df.]	pivot_table	e(index='Name',values='Marks',aggfunc='mean')
		or
>>> pv=df	.pivot_ta	able(index='Name',aggfunc='mean')
>>> pv		
	Marks	
Name		
Aryan	66.5	
Pratibha	97.0	
Ritika	85.5	
Saumya	86.5	

pivot table can be created by the following method also. \rightarrow

>>> pd.pivo	<pre>ot_table(df,index='Name',aggfunc='mean')</pre>
Ν	larks
Name	
Aryan	66.5
Pratibha	97.0
Ritika	85.5
Saumya	86.5

>>> pv=df	.pivot	table	(index=	= ' Nam	le '	, CO	lum	ns='Subject',aggfunc='mean')
>>> pv								
	Marks							
Subject	CS	IP			Pay	/ atte	ntic	on on the values of aggfunc
Name				<u> </u>		_]_] _ (.		
Aryan	64.5	68.5	>>> pv=a >>> pv		L_Li	abie (.	Inde	x='Name', columns='Subject', agglunc='count')
Pratibha	98.0	96.0		Marks		Test		
Ritika	86.0	85.0	Subject Name	CS	ΙP	CS	ΙP	
Saumya	88.5	84.5	Aryan	2	2	2	2	
· · · · · · · · · · · · · · · · · · ·			Pratibha	2	2	2	2	
			Ritika	2	2	2	2	
			Saumya	2	2	2	2	

Neha Tyagi, KV No-5 Jaipur

An excercise for you \rightarrow

An Emp table contains the following data:

Empno	Name	Department	Salary	Commission	dof
100	Sunita Sharma	RESEARCH	45600	5600.0	CLERK
101	Ashok Singhal	SALES	43900	3900.0	SALESMAN
102	Sumit Avasti	SALES	27000	7000.0	SALESMAN
103	Jyoti Lamba	RESEARCH	45900	4900.0	MANAGER
104	Martin S.	SALES	32500	3500.0	SALESMAN
105	Binod Goel	SALES	45200	4200.0	MANAGER
106	Chetan Gupta	ACCOUNTS	36800	6800.0	MANAGER
107	Sudhir Rawat	RESEARCH	37000	7000.0	ANALYST
108	Kavita Sharma	ACCOUNTS	42900	4900.0	CLERK
109	Tushar Tiwari	SALES	49500	4500.0	MANAGER
110	Anand Rathi	OPERATIONS	41600	8200.0	SR. MANAGER
111	Sumit Vats	RESEARCH	47800	NaN	SR. MANAGER
112	Manoj Kaushik	OPERATIONS	43600	NaN	CLERK

- (a) Using above table create a DataFrame called dfE.
- (b) Display the department wise total salary.
- (c) Display the department wise average salary.
- (d) Display the department wise total and average salary.
- (e) Display the department wise maximum and minimum salary.
- (f) Display the department and job wise maximum salary.

Pivoting using pivot_table() method... Solution→

First you will create a dataframe of table using pandas.

After that you have to apply the following functions.-

- (b) pd.pivot_table(dfE, index='Department', values='Salary', aggfunc='sum')
- (c) pd.pivot_table(dfE, index='Department', values='Salary') Or

pd.pivot_table(dfE, index='Department', values='Salary', aggfunc='mean')

- (d) pd.pivot_table(dfE, index='Department', values='Salary', aggfunc=['sum', 'mean'])
- (e) pd.pivot_table(dfE, index='Department', values='Salary', aggfunc=['max', 'min'])
- (f) pd.pivot_table(dfE, index=['Department', 'Job'], values='Salary', aggfunc='max')

Sorting of DataFrames

- Data of DataFrame can be sort according to values of row and column.
- By default sorting is done on row labels in ascending order.
- Pandas DataFrames has two useful sort functions \rightarrow
 - *sort_values():* it sorts the data of given column to the function in ascending or descending order.
 - *sort_index():* this function sorts rows (axis=0) or columns (axis=1).
- Its syntax is as follows →
- DataFrame.sort_values(by = None, axis=0, ascending = True, inplace = False)
- DataFrame.sort_index(by = None, axis=0, ascending = True, inplace = False)
- Here
 - by: column to be sorted.
 - axis: here passing 0 means sorting will be done row wise and 1 means column wise
 - ascending: by default ascending is true
 - inplace: default is false if you don't want a new dataframe then set it true.

DataFrames Sorting...

>>	→> df				>>	> dfn=df.	sort_val	.ues('N	ame')
	Name	Subject	Marks	Grad	le 🔚	or			
0	Pratibha	CS	99	A	\+	VI dfn-df	ant valu	loc (bu-	Namo
1	Ritika	IP	87		A	> din-ui.: > dfn	SOLL_VALU	les (by–	Name
2	Saumya	PHY	88	В	3+ / _	> uin Namo	Subject	Marke	Crado
3	Aryan	CHEM	67		B 3	Aryan	CHEM	Marks 67	GLAUE
by	default sorting i	s in ascending	g order.			Dratibba	CILEM	907	Д Д
					1	Ritika	ТР	87	
	sort in desceidir	ng order the ex	xample is as	s under.	2	Saumva	рну	88	R+
	dfn-df a	ort volu	oc (by-	Namo		nding-Fa		00	
~~	/ uin-ui.s	OIL_VAIU	les (by-	Manie	, asce	enurny-ra.			
//	> uin	Cubicat	Monleg	Cread	~		N		
2	Name	Subject	Marks	Grad	.e		1		
Z 1	Saumya	PHI	88	В	+ 7	Value o	of Ascending	parameter i	S
T	RITIKA	1 P	87		A	false			
0	Pratibha	CS	99	A	.+ 				
3	Aryan	CHEM	67		В				
>>	>> dfn=df	.sort_v	alues(['Nai	me ','	Marks'],	ascendi	ng=Tru	ie)
>>	>> dfn								
	Nam	e Subje	ct Ma	rks	Grade	If we a	ive two col	umns like	2
3	Arva	n CHI	EM	67	В	this the	n sorting o	n multin	
0	Pratibh	a (CS	99	Δ+		a will be d	ono	
1		u .		22 07	771	Column		one.	
L L	KIUIK ~	a .	T F.	0 /	A				
2	Saumy	a Pl	ΗY	88	B+				

Sort by index

		-										
>>:	> dfn=df.s	sort_inde	x()		>>>	>>> dfn=df.sort_index(ascending=False						
>>:	> dfn				>>>	> dfn	—					
	Name	Subject	Marks	Grade		Name	Subject	Marks	Grade			
0	Pratibha	CS	99	A+	3	Aryan	CHEM	67	В			
1	Ritika	IP	87	А	2	Saumya	PHY	88	B+			
2	Saumya	PHY	88	B+	1	Ritika	IP	87	А			
3	Aryan	CHEM	67	В	0	Pratibha	CS	99	A+			
	Sorting i	n ascendir	ng order			Sortin	g in desce	nding or	der			

Points to remember:

- 1. pivot() method creates a new table whose row and column are unique.
- 2. pivot() method is used to pivot without aggregation.
- *3. stacking* means moving innermost column index to innermost row index.

• Please follow our blog and subscribe youtube channel to get all the chapter and lectures.

www.pythontrends.wordpress.com

एक शुरुआत pythontrends

पाइथन सीखें और सिखाएं

मुख्य पृष्ठ/Home

संपर्क/Contact

कक्षा-11 आई० पी० /Class -XI IP 🗸

कक्षा-11 कंप्यूटर साइंस/Class -XI Computer Science

कक्षा -12 कंप्यूटर साइंस/Class-12 CS

~

पाइथन प्रोग्राम और SQL कनेक्टिविटी / Python Program and SQL connectivity

कार्य /Assignments

पाठ्यक्रम(CS और IP)/syllabus(CS and IP)

नमस्ते दोस्तों ! /Hello Friends!



यह ब्लॉग उन बच्चों की मदद के लिए बनाया गया है जो python में प्रोग्रामिंग सीख रहे हैं | यह ब्लॉग द्विभाषीय होगा जिससे सीबीएसई बोर्ड के वे बच्चे जिन्हें अंग्रजी भाषा में समस्या होती है उन्हें सही मार्गदर्शन करेगा तथा प्रोग्रामिंग में उनकी सहायता करेगा | जैसा की हम जानते हैं की हमारे देश में कई क्षेत्र और कई लोग ऐसे हैं जिनकी अंग्रेज़ी उतनी मज़बूत नहीं है क्यों कि ये हमारी मातृभाषा नहीं है | तो हमें कभी कभी अंग्रेज़ी के कठिन शब्दों को समझने में समय लगता है और ये समय अगर लॉजिकल विचारों में लगे तो छात्रों का अधिक भला हो सकता है | इस ब्लॉग पर हम कोशिश करेंगे की पाइथन से सम्बंधित सभी तथ्य तथा सामग्री इस ब्लॉग पर उपलब्ध कराएं | यह ब्लॉग **संजीव भदौरिया (पी जी टी कंप्यूटर साइंस) के० वि० वारावंकी लखनऊ संभाग एवं नेहा त्यागी (पी जी टी कंप्यूटर साइंस) के० वि० क्रं -5 जयपुर,**