



pandas

Python



Pandas



ANACONDA



SANTOSH NAUPADA

The screenshot displays a Jupyter Notebook titled 'pandas codinggrad' with a last checkpoint from yesterday at 2:23 PM. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, undo, redo, and running code. The notebook contains several code cells and their corresponding outputs.

Cell 2: Imports numpy as np and pandas as pd.

```
In [2]: 1 import numpy as np
        2 import pandas as pd
```

Cell 3: Creates a pandas Series 's1' with values [45, 48, 47, 3, 25, 66].

```
In [3]: 1 #a pandas series is column in table
        2 # it is 1d array holding data in any
        3 s1=pd.Series([45,48,47,3,25,66])
        4 s1
```

Output 3: A pandas Series with index 0 to 5 and dtype int64.

```
Out[3]: 0    45
        1    48
        2    47
        3     3
        4    25
        5    66
        dtype: int64
```

Cell 4: Creates a pandas Series 's1' with values [45, 48, 47, 3, 25, 66] and index ['a', 'b', 'c', 'd', 'e', 'f'].

```
In [4]: 1 s1=pd.Series([45,48,47,3,25,66],index=['a','b','c','d','e','f'])
        2 s1
```

Output 4: A pandas Series with index 'a' to 'f' and dtype int64.

```
Out[4]: a    45
        b    48
        c    47
        d     3
        e    25
        f    66
        dtype: int64
```

Cell 5: Creates a pandas Series 's1' with values [45, 48, 47, 3, 25, 66] and index ['a', 'b', 'c', 'd', 'e', 'f'], dtype float.

```
In [5]: 1 s1=pd.Series([45,48,47,3,25,66],index=['a','b','c','d','e','f'],dtype=float)
        2 s1
        3
```

Output 5: A pandas Series with index 'a' to 'f' and dtype float64.

```
Out[5]: a    45.0
        b    48.0
        c    47.0
        d     3.0
        e    25.0
        f    66.0
        dtype: float64
```

Cell 7: Accesses the value at index 'd' of series 's1'.

```
In [7]: 1 s1['d']
```

Output 7: 3.0

```
Out[7]: 3.0
```

Cell 8: A comment stating 'Series is 1-d array, DataFrame-2d array'.

```
In [8]: 1 ###Series is 1-d array, DataFrame-2d array
```

Cell 9: Creates a pandas Series 's2' with values {a: 228, b: 885, c: 452}.

```
In [9]: 1 s2=pd.Series({'a':228,'b':885,'c':452})
        2 s2
```

Output 9: A pandas Series with index 'a' to 'c' and dtype int64.

```
Out[9]: a    228
        b    885
        c    452
        dtype: int64
```

Cell 10: Converts a list of values [245, 25, 26, 28] into a DataFrame 'd1'.

```
In [10]: 1 # convert series into dataframe
        2 d1=pd.DataFrame([245,25,26,28])
        3 d1
```

Output 10: A DataFrame with index 0 to 3 and dtype int64.

```
Out[10]:   0
0  245
1   25
2   26
3   28
```

Cell 11: Creates a DataFrame 'd2' with values [[245, 25, 26, 28], [36, 35, 85, 44], [55, 666, 88, 22]].

```
In [11]: 1 d2=pd.DataFrame([[245,25,26,28],[36,35,85,44],[55,666,88,22]])
        2 d2
```

Output 11: A DataFrame with index 0 to 2 and dtype int64.

```
Out[11]:   0  1  2  3
0  245  25  26  28
1   36  35  85  44
2   55 666  88  22
```

jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [12]:

```
1 d2=pd.DataFrame([[245,25,26,28],[36,35,85,44],[55,666,88,22]],columns=['a','b','c','d'])
2 d2
```

Out[12]:

	a	b	c	d
0	245	25	26	28
1	36	35	85	44
2	55	666	88	22

In [13]:

```
1 d2=pd.DataFrame([[245,25,26,28],[36,35,85,44],[55,666,88,22]],columns=['a','b','c','d'],index=['x','y','z'])
2 d2
```

Out[13]:

	a	b	c	d
x	245	25	26	28
y	36	35	85	44
z	55	666	88	22

In [14]:

```
1 ## diff btwn dataframe and series
2 d3=pd.DataFrame([{'sun':11,'mon':22,'tues':33},{ 'wed':44,'thur':55,'fri':66,'sat':77}],index=['x','y'])
3 d3
```

Out[14]:

	sun	mon	tues	wed	thur	fri	sat
x	11.0	22.0	33.0	NaN	NaN	NaN	NaN
y	NaN	NaN	NaN	44.0	55.0	66.0	77.0

In [15]:

```
1 d4=pd.Series([{'sun':11,'mon':22,'tues':33},{ 'wed':44,'thur':55,"fri":66,'sat':77}],index=['x','y'])
2 d4
```

Out[15]:

```
x      {'sun': 11, 'mon': 22, 'tues': 33}
y      {'wed': 44, 'thur': 55, 'fri': 66, 'sat': 77}
dtype: object
```

jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [16]:

```
1 d2
```

Out[16]:

	a	b	c	d
x	245	25	26	28
y	36	35	85	44
z	55	666	88	22

In [17]:

```
1 d2['c']
```

Out[17]:

```
x      26
y      85
z      88
Name: c, dtype: int64
```

In [18]:

```
1 d2['e']=d2['a']+d2['b']
2 d2['f']=d2['c']+d2['d']
3 d2
```

Out[18]:

	a	b	c	d	e	f
x	245	25	26	28	270	54
y	36	35	85	44	71	129
z	55	666	88	22	721	110

In [19]:

```
1 del d2['e']
2 d2
```

Out[19]:

	a	b	c	d	f
x	245	25	26	28	54
y	36	35	85	44	129
z	55	666	88	22	110

jupyter pandas codinggrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3

```
In [20]: 1 d2.insert(2,'new',d2['a'])
         2 d2
```

```
Out[20]:
```

	a	b	new	c	d	f
x	245	25	245	26	28	54
y	36	35	36	85	44	129
z	55	666	55	88	22	110

```
In [21]: 1 d2.info()

<class 'pandas.core.frame.DataFrame'>
Index: 3 entries, x to z
Data columns (total 6 columns):
#   Column  Non-Null Count  Dtype  
---  -
0    a         3 non-null    int64  
1    b         3 non-null    int64  
2   new         3 non-null    int64  
3    c         3 non-null    int64  
4    d         3 non-null    int64  
5    f         3 non-null    int64  
dtypes: int64(6)
memory usage: 276.0+ bytes
```

```
In [22]: 1 p1=pd.DataFrame({'col1':[1,2,3,4,5,6,7,8,9],
         2                  'col2':[11,22,33,44,55,66,77,88,99],
         3                  'col3':['a','b','c','d','e','f','g','h','i']
         4                  })
         5 p1
```

```
Out[22]:
```

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c

The image shows a Jupyter Notebook interface with the title 'pandas codingrad'. The top bar indicates the last checkpoint was 'Yesterday at 2:23 PM (autosaved)'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, running, and other actions. The notebook contains three code cells:

In [22]:

```
1 p1=pd.DataFrame({'col1':[1,2,3,4,5,6,7,8,9],
2                  'col2':[11,22,33,44,55,66,77,88,99],
3                  'col3':['a','b','c','d','e','f','g','h','i']
4                  })
5 p1
```

Out[22]:

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c
3	4	44	d
4	5	55	e
5	6	66	f
6	7	77	g
7	8	88	h
8	9	99	i

In [23]:

```
1 p1.head(5)
```

Out[23]:

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c
3	4	44	d
4	5	55	e

In [24]:

```
1 p1.tail(3)
```

The image shows a JupyterLab interface with a pandas DataFrame named 'p1'. The DataFrame has three columns: 'col1', 'col2', and 'col3'. The interface displays the results of three code cells:

Cell 24: `p1.tail(3)`

	col1	col2	col3
6	7	77	g
7	8	88	h
8	9	99	i

Cell 25: `p1`

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c
3	4	44	d
4	5	55	e
5	6	66	f
6	7	77	g
7	8	88	h
8	9	99	i

Cell 26: `p1.loc[6, 'col2']` # is loc[rows, columns]

77

Cell 27: `p1.loc[3:7, ['col2', 'col3']]`

	col2	col3
3	44	d
4	55	e

Jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python

Out[26]: //

In [27]: 1 p1.loc[3:7,['col2','col3']]

Out[27]:

	col2	col3
3	44	d
4	55	e
5	66	f
6	77	g
7	88	h

In [28]: 1 p1.loc[2:6:8,['col1','col3']]

Out[28]:

	col1	col3
2	3	c

In [29]: 1 p1.loc[[3,7,4],['col2','col3']]

Out[29]:

	col2	col3
3	44	d
7	88	h
4	55	e

In [30]: 1 p1

Out[30]:

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c

jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted

In [30]: 1 p1

Out[30]:

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c
3	4	44	d
4	5	55	e
5	6	66	f
6	7	77	g
7	8	88	h
8	9	99	i

In [31]: 1 p1.iloc[5,1] *# iloc=not mention col names, just give row, col indexes*

Out[31]: 66

In [32]: 1 p1.iloc[2:6,0:2]

Out[32]:

	col1	col2
2	3	33
3	4	44
4	5	55
5	6	66

In [33]: 1 p1.col2

Out[33]:

	col2
0	11
1	22
2	33
3	44
4	55
5	66
6	77
7	88
8	99

Name: col2, dtype: int64

In [33]: 1 p1.col2

Out[33]:

	col2
0	11
1	22
2	33
3	44
4	55
5	66
6	77
7	88
8	99

Name: col2, dtype: int64

In [34]: 1 p1['col2']

Out[34]:

	col2
0	11
1	22
2	33
3	44
4	55
5	66
6	77
7	88
8	99

Name: col2, dtype: int64

In [35]: 1 p1.col3.values

Out[35]: array(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i'], dtype=object)

In [36]: 1 p1.sum

Out[36]:

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c
3	4	44	d
4	5	55	e
5	6	66	f

Cell 37:

```
In [37]: 1 p1
```

```
Out[37]:
```

	col1	col2	col3
0	1	11	a
1	2	22	b
2	3	33	c
3	4	44	d
4	5	55	e
5	6	66	f
6	7	77	g
7	8	88	h
8	9	99	i

Cell 38:

```
In [38]: 1 p2=pd.DataFrame({'name':['john','kane','kate','karan'],
2                        'id':['25','35','45','75'],
3                        'profit':['12500','22500','8800','52500']})
4 p2
```

```
Out[38]:
```

	name	id	profit
0	john	25	12500
1	kane	35	22500
2	kate	45	8800
3	karan	75	52500

Cell 39:

```
In [39]: 1 a=[['john',25,'12500'],['kane',35,'22500'],['kane',45,'8800'],['karan',75,'52500']]
2 p2=pd.DataFrame(a,columns=['name','id','profit'])
3 p2
```

```
Out[39]:
```

	name	id	profit
0	john	25	12500
1	kane	35	22500
2	kane	45	8800
3	karan	75	52500

Cell 40:

```
In [40]: 1 p2.append({'name':'honey','id':75,'profit':32500},ignore_index=True)
```

```
Out[40]:
```

	name	id	profit
0	john	25	12500
1	kane	35	22500
2	kane	45	8800
3	karan	75	52500
4	honey	75	32500

Cell 41:

```
In [41]: 1 import keyword
2 print(keyword.kwlist)
```

```
Out[41]:
```

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Cell 42:












```
In [42]: 1 p3=p2.append({'name':np.nan,'id':78,'profit':192500},ignore_index=True)
2 p3
```

```
Out[42]:
```

	name	id	profit
0	john	25	12500
1	kane	35	22500
2	kane	45	8800
3	karan	75	52500
4	honey	75	32500
5	NaN	78	192500

jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 C

In [42]:

```
1 p3=p2.append({'name':np.nan,'id':78,'profit':192500},ignore_index=True)
2 p3
```

Out[42]:

	name	id	profit
0	john	25.0	12500
1	kane	35.0	22500
2	kane	45.0	8800
3	karan	75.0	52500
4	NaN	78.0	192500.0

In [43]:

```
1 p3.isnull()
```

Out[43]:

	name	id	profit
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	True	False	False

In [44]:

```
1 p3.isnull().sum()
```

Out[44]:

```
name      1
id         0
profit     0
dtype: int64
```

In [45]:












```
1 p3.dropna()
```

Out[45]:

	name	id	profit
0	john	25.0	12500
1	kane	35.0	22500
2	kane	45.0	8800
3	karan	75.0	52500

jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 O

In [45]:

```
1 p3.dropna()
```

Out[45]:

	name	id	profit
0	john	25.0	12500
1	kane	35.0	22500
2	kane	45.0	8800
3	karan	75.0	52500

In [46]:

```
1 p3.fillna(value='fun')
```

Out[46]:

	name	id	profit
0	john	25.0	12500
1	kane	35.0	22500
2	kane	45.0	8800
3	karan	75.0	52500
4	fun	78.0	192500.0

In [47]:

```
1 p4=pd.DataFrame({'fruits':['apple','mango','apple','mango','grapes'],'price':[200,120,80,100,90] })
2 p4
```

Out[47]:

	fruits	price
0	apple	200
1	mango	120
2	apple	80
3	mango	100
4	grapes	90

Jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [56]: 1 p4.groupby('fruits')
Out[56]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000239D8ECD10>

In [48]: 1 p4.groupby(['fruits']).mean()
Out[48]:
```

	price
fruits	
apple	140
grapes	90
mango	110

```
In [49]: 1 p4
Out[49]:
```

	fruits	price
0	apple	200
1	mango	120
2	apple	80
3	mango	100
4	grapes	90

```
In [50]: 1 data1=pd.read_csv('snapdeal-audio-webscrapping_data.csv')
2 data1
Out[50]:
```

	Unnamed: 0	audio_names	orinal_price	offer_price	discount	images
0	0	iGear Ensemble 20 watts Wooden Portable Soundb...	Rs. 3,000	Rs. 1,999	33% Off	space
6 for key,value in data2.iteritems():
7
8 print(key,value)
9 print('\n')
```

|   | col1     | col2     | col3     | col4     | col5     |
|---|----------|----------|----------|----------|----------|
| 0 | 0.918603 | 0.355455 | 0.006694 | 0.262372 | 0.968961 |
| 1 | 0.956700 | 0.883580 | 0.224644 | 0.138874 | 0.953598 |
| 2 | 0.545801 | 0.216136 | 0.655636 | 0.576543 | 0.245625 |
| 3 | 0.531401 | 0.439075 | 0.173514 | 0.257874 | 0.354300 |

```
col1 0 0.918603
1 0.956700
2 0.545801
3 0.531401
```

The image shows a Jupyter Notebook interface with the title "pandas codingrad". The top bar indicates the last checkpoint was yesterday at 2:23 PM (autosaved). The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running code. The notebook content displays the output of several pandas operations:

```

1 0.883580
2 0.216136
3 0.439075
Name: col2, dtype: float64

col3 0 0.006694
1 0.224644
2 0.655636
3 0.173514
Name: col3, dtype: float64

col4 0 0.262372
1 0.138874
2 0.576543
3 0.257874
Name: col4, dtype: float64

col5 0 0.968961
1 0.953598
2 0.245625
3 0.354300
Name: col5, dtype: float64

```

Below the output, a code cell (In [84]:) contains the following Python code:

```

1 data2=pd.DataFrame(np.random.rand(4,5),columns=['col1','col2','col3','col4','col5'])
2 print(data2)
3
4 print('\n') # for
5 # for \n=> space
6 for row in data2.iterrows():
7
8 print(row)
9 print('\n')

```

The final output of the code cell is a pandas DataFrame:

|   | col1     | col2     | col3     | col4     | col5     |
|---|----------|----------|----------|----------|----------|
| 0 | 0.664451 | 0.957311 | 0.604386 | 0.015602 | 0.268755 |

jupyter pandas codinggrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 O

(1, col1 0.574981  
 col2 0.961705  
 col3 0.586519  
 col4 0.951837  
 col5 0.009044  
 Name: 1, dtype: float64)

(2, col1 0.375636  
 col2 0.467468  
 col3 0.574920  
 col4 0.532862  
 col5 0.880389  
 Name: 2, dtype: float64)

(3, col1 0.239151  
 col2 0.461321  
 col3 0.892838  
 col4 0.823261  
 col5 0.510865  
 Name: 3, dtype: float64)

In [87]: 1 data2.T # transpose:: rows=col,col=rows

Out[87]:

|      | 0        | 1        | 2        | 3        |
|------|----------|----------|----------|----------|
| col1 | 0.664451 | 0.574981 | 0.375636 | 0.239151 |
| col2 | 0.957311 | 0.961705 | 0.467468 | 0.461321 |
| col3 | 0.604386 | 0.586519 | 0.574920 | 0.892838 |
| col4 | 0.015602 | 0.951837 | 0.532862 | 0.823261 |
| col5 | 0.268755 | 0.009044 | 0.880389 | 0.510865 |

jupyter pandas codinggrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run

```
In [98]: 1 a=(['arun','vizag','225','A.P'],['varun','badrak','226','ORS'],
2 ['kiran','kolkata','227','WB'],['anand','salem','228','TN'])
3
4 data3 = pd.DataFrame(a,columns=['name','city','employee','state'])
5 data3
```

```
Out[98]:
```

|   | name  | city    | employee | state |
|---|-------|---------|----------|-------|
| 0 | arun  | vizag   | 225      | A.P   |
| 1 | varun | badrak  | 226      | ORS   |
| 2 | kiran | kolkata | 227      | WB    |
| 3 | anand | salem   | 228      | TN    |

```
In [99]: 1 pd.get_dummies(data3['employee'])
```

```
Out[99]:
```

|   | 225 | 226 | 227 | 228 |
|---|-----|-----|-----|-----|
| 0 | 1   | 0   | 0   | 0   |
| 1 | 0   | 1   | 0   | 0   |
| 2 | 0   | 0   | 1   | 0   |
| 3 | 0   | 0   | 0   | 1   |

```
In [100]: 1 pd.get_dummies(data3['name'])
```

```
Out[100]:
```

|   | anand | arun | kiran | varun |
|---|-------|------|-------|-------|
| 0 | 0     | 1    | 0     | 0     |
| 1 | 0     | 0    | 0     | 1     |
| 2 | 0     | 0    | 1     | 0     |
| 3 | 1     | 0    | 0     | 0     |









jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3

In [107]:

```
1 frames=[data1,data2,data3]
2 result=pd.concat(frames,axis=1)
3 result
```

Out[107]:

| Unnamed: 0 | audio_names                                       | orinal_price | offer_price | discount | images                                                                                                                            | col1     | col2     | col3     | col4     | col5     | name  | city    | employee |
|------------|---------------------------------------------------|--------------|-------------|----------|-----------------------------------------------------------------------------------------------------------------------------------|----------|----------|----------|----------|----------|-------|---------|----------|
| 0          | iGear Ensemble 20 watts Wooden Portable Soundb... | Rs. 3,000    | Rs. 1,999   | 33% Off  |  class="product-image" src="https://n4.sdl...  | 0.664451 | 0.957311 | 0.604386 | 0.015602 | 0.268755 | arun  | vizag   | 225      |
| 1          | iGear Delight 10 Watts Wireless Sound bar Spea... | Rs. 1,999    | Rs. 1,750   | 12% Off  |  class="product-image" src="https://n4.sdl...  | 0.574981 | 0.961705 | 0.586519 | 0.951837 | 0.009044 | varun | badrak  | 226      |
| 2          | Landmark ROAR Soundbar                            | Rs. 2,199    | Rs. 2,009   | 9% Off   |  class="product-image" src="https://n4.sdl...  | 0.375636 | 0.467468 | 0.574920 | 0.532862 | 0.880389 | kiran | kolkata | 227      |
| 3          | Landmark GOD OF BASS Soundbar                     | Rs. 3,999    | Rs. 2,885   | 17% Off  |  class="product-image" src="https://n2.sdl...  | 0.239151 | 0.461321 | 0.892838 | 0.823261 | 0.510865 | anand | salem   | 228      |
| 4          | Landmark ROAR Soundbar                            | Rs. 5,999    | Rs. 1,075   | 9% Off   |  class="product-image lazy-load" data-src=...  | NaN      | NaN      | NaN      | NaN      | NaN      | NaN   | NaN     | NaN      |
| 5          | Landmark Carrier Hi-Fi System                     | Rs. 4,999    | Rs. 2,009   | 28% Off  |  class="product-image lazy-load" data-src=...  | NaN      | NaN      | NaN      | NaN      | NaN      | NaN   | NaN     | NaN      |
| 6          | Landmark Volcano Hi-Fi System                     | Rs. 2,199    | Rs. 2,009   | 17% Off  |  class="product-image lazy-load" data-src=... | NaN      | NaN      | NaN      | NaN      | NaN      | NaN   | NaN     | NaN      |
| 7          | Landmark Pump Hi-Fi                               | Rs. 1,299    | Rs. 2,996   | 40% Off  |  class="product-image lazy-load"             | NaN      | NaN      | NaN      | NaN      | NaN      | NaN   | NaN     | NaN      |



jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

+ ⌕ 📄 📁 ⬆ ⬇ ▶ Run ⏹ 🔄 ⏩ Code ⌵ 🗨

```
In [115]: 1 dat1={
2 'name':['moon','sun','mars'],
3 'distance':['81ak','120lak','65lak']
4 }
5 dat2={
6 'name':['saturn','sun','jupiter'],
7 'distance':['440lak','220lak','320lak']
8 }
9 DF1=pd.DataFrame(dat1)
10 DF2=pd.DataFrame(dat2)
11
12 newdf=DF1.merge(DF2,how='right')
13 newdf
```

Out[115]:

|   | name    | distance |
|---|---------|----------|
| 0 | saturn  | 440lak   |
| 1 | sun     | 220lak   |
| 2 | jupiter | 320lak   |

```
In [116]: 1 DF3=pd.merge(DF1,DF2,on='name',how='outer')
2 DF3
```

Out[116]:

|   | name    | distance_x | distance_y |
|---|---------|------------|------------|
| 0 | moon    | 81ak       | NaN        |
| 1 | sun     | 120lak     | 220lak     |
| 2 | mars    | 65lak      | NaN        |
| 3 | saturn  | NaN        | 440lak     |
| 4 | jupiter | NaN        | 320lak     |

```
In [117]: 1 DF3=pd.merge(DF1,DF2,on='name',how='right')
2 DF3
```

**Jupyter pandas codinggrad** Last Checkpoint: Yesterday at 2:23 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [117]:

```
1 DF3=pd.merge(DF1,DF2,on='name',how='right')
2 DF3
```

Out[117]:

|   | name    | distance_x | distance_y |
|---|---------|------------|------------|
| 0 | saturn  | NaN        | 440lak     |
| 1 | sun     | 120lak     | 220lak     |
| 2 | jupiter | NaN        | 320lak     |

In [118]:

```
1 DF3=pd.merge(DF1,DF2,on='name',how='left')
2 DF3
```

Out[118]:

|   | name | distance_x | distance_y |
|---|------|------------|------------|
| 0 | moon | 8lak       | NaN        |
| 1 | sun  | 120lak     | 220lak     |
| 2 | mars | 65lak      | NaN        |

In [122]:

```
1 weather=pd.DataFrame({
2 'city':['vizag','kurnool','ongole'],
3 'humidity':['25','30','32'],
4 'temperature':['30deg','32deg','33deg'],
5 })
6 weather
```

Out[122]:

|   | city    | humidity | temperature |
|---|---------|----------|-------------|
| 0 | vizag   | 25       | 30deg       |
| 1 | kurnool | 30       | 32deg       |
| 2 | ongole  | 32       | 33deg       |

In [123]:

```
1 weather1=pd.DataFrame({
2 'city':['aus','nz','pak'],
3 'humidity':['22','20','22'],
4 'temperature':['16deg','12deg','30deg'],
5 })
6 weather1
```

Out[123]:

|   | city | humidity | temperature |
|---|------|----------|-------------|
| 0 | aus  | 22       | 16deg       |
| 1 | nz   | 20       | 12deg       |
| 2 | pak  | 22       | 30deg       |

In [125]:

```
1 x=pd.concat([weather,weather1])
2 x
```

Out[125]:

|   | city    | humidity | temperature |
|---|---------|----------|-------------|
| 0 | vizag   | 25       | 30deg       |
| 1 | kurnool | 30       | 32deg       |
| 2 | ongole  | 32       | 33deg       |
| 0 | aus     | 22       | 16deg       |
| 1 | nz      | 20       | 12deg       |
| 2 | pak     | 22       | 30deg       |

jupyter pandas codingrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [127]:

```
1 x=pd.concat([weather,weather1],ignore_index=True)
2 x
```

Out[127]:

|   | city    | humidity | temperature |
|---|---------|----------|-------------|
| 0 | vizag   | 25       | 30deg       |
| 1 | kurnool | 30       | 32deg       |
| 2 | ongole  | 32       | 33deg       |
| 3 | aus     | 22       | 16deg       |
| 4 | nz      | 20       | 12deg       |
| 5 | pak     | 22       | 30deg       |

In [133]:

```
1 x=pd.concat([weather,weather1],keys=['aus','pak'])
2 x
```

Out[133]:

|       | city    | humidity | temperature |
|-------|---------|----------|-------------|
| aus 0 | vizag   | 25       | 30deg       |
| 1     | kurnool | 30       | 32deg       |
| 2     | ongole  | 32       | 33deg       |
| pak 0 | aus     | 22       | 16deg       |
| 1     | nz      | 20       | 12deg       |
| 2     | pak     | 22       | 30deg       |

In [142]:

```
1 weather
```

Out[142]:

|   | city    | humidity | temperature |
|---|---------|----------|-------------|
| 0 | vizag   | 25       | 30deg       |
| 1 | kurnool | 30       | 32deg       |
| 2 | ongole  | 32       | 33deg       |

jupyter pandas codinggrad Last Checkpoint: Yesterday at 2:23 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 C

Run

```

aus 0 vizag 25 30deg
 1 kurnool 30 32deg
 2 ongole 32 33deg
pak 0 aus 22 16deg
 1 nz 20 12deg
 2 pak 22 30deg

```

In [142]: 1 weather

Out[142]:

|   | city    | humidity | temperature |
|---|---------|----------|-------------|
| 0 | vizag   | 25       | 30deg       |
| 1 | kurnool | 30       | 32deg       |
| 2 | ongole  | 32       | 33deg       |

In [148]: 1 weather.corr # co-relation

Out[148]: <bound method DataFrame.corr of

|   | city    | humidity | temperature |
|---|---------|----------|-------------|
| 0 | vizag   | 25       | 30deg       |
| 1 | kurnool | 30       | 32deg       |
| 2 | ongole  | 32       | 33deg       |

In [ ]: 1