Objective: learn to create dataframe and apply join operations between dataframes

1)Concatenating 2)Append 3)Merge

In [2]:

```python
# dataframe 1
import pandas as pd

# assign data of lists.
data = {'city': ['delhi', 'mumbai', 'agra', 'goa'], \
        'positive': [20, 21, 19, 18],'neagtive': [120, 121, 119, 18] }

# Create DataFrame
df1 = pd.DataFrame(data)
```

In [3]:

```python
df1
```

Out[3]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119
3	goa	18	18

In [4]:

```python
# dataframe 2
data = {'city': ['delhi', 'mumbai', 'agra', 'chennai'], \
        'positive': [10, 21, 39, 18],'neagtive': [12, 101, 129, 118] }

# Create DataFrame
df2 = pd.DataFrame(data)
df2
```

Out[4]:

	city	positive	neagtive
0	delhi	10	12
1	mumbai	21	101
2	agra	39	129
3	chennai	18	118

In [99]:

```python
# concatenate: concatenate the two dataframes one below the other.
df3 = pd.concat([df1,df2])
```

```
In [100]: df3
```

Out[100]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119
3	goa	18	18
0	delhi	10	12
1	mumbai	21	101
2	agra	39	129
3	chennai	18	118

We see that in above result, we did not get continuous indexes(0,1,2,3,0,1,2,3) to make them continuous like 0,1,2,3,4,... we can write ignore_index=True

```
In [101]: df3 = pd.concat([df1,df2], ignore_index=True)
```

```
In [102]: df3
```

Out[102]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119
3	goa	18	18
4	delhi	10	12
5	mumbai	21	101
6	agra	39	129
7	chennai	18	118

Assignng keys to dataframes df1 and df2

```
In [103]: df3 = pd.concat([df1,df2], keys = ['first', 'second'])
```

In [104]:

```python
df3
```

Out[104]:

		city	positive	neagtive
first	**0**	delhi	20	120
	1	mumbai	21	121
	2	agra	19	119
	3	goa	18	18
second	**0**	delhi	10	12
	1	mumbai	21	101
	2	agra	39	129
	3	chennai	18	118

In [105]:

```python
df3.loc['first']
```

Out[105]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119
3	goa	18	18

In [106]:

```python
df3.loc['first', 0]
```

Out[106]:

```
city         delhi
positive        20
neagtive       120
Name: (first, 0), dtype: object
```

In [107]:

```python
df3.loc['second']
```

Out[107]:

	city	positive	neagtive
0	delhi	10	12
1	mumbai	21	101
2	agra	39	129
3	chennai	18	118

In [108]:

```python
# if you want to combine the two data frames horizontally means one nex
df3 = pd.concat([df1,df2], axis =1)
df3
```

Out[108]:

	city	positive	neagtive	city	positive	neagtive
0	delhi	20	120	delhi	10	12
1	mumbai	21	121	mumbai	21	101
2	agra	19	119	agra	39	129
3	goa	18	18	chennai	18	118

Another example: Create two dataframes and concatenate them horizontally (axis =1)

In [5]:

```
# dataframe 1
import pandas as pd

# assign data of lists.
data = {'city': ['delhi', 'mumbai', 'agra', 'goa'], \
        'temperature': [20, 21, 19, 18]}

# Create DataFrame
df1 = pd.DataFrame(data)
```

In [110]:

```
df1
```

Out[110]:

```
     city  temperature
0   delhi           20
1  mumbai           21
2    agra           19
3     goa           18
```

In [6]:

```
# dataframe 2
import pandas as pd

# assign data of lists.
data = {'city': ['agra','mumbai','goa','delhi',], \
        'windspeed': [2, 2, 1, 1]}

# Create DataFrame
df2 = pd.DataFrame(data)
```

In [112]:

```
df2
```

Out[112]:

```
     city  windspeed
0    agra          2
1  mumbai          2
2     goa          1
3   delhi          1
```

In [113]:

```
df3 = pd.concat([df1,df2], axis =1)
df3
```

Out[113]:

```
     city  temperature    city  windspeed
0   delhi           20    agra          2
1  mumbai           21  mumbai          2
2    agra           19     goa          1
3     goa           18   delhi          1
```

We see in the above output the rows are not containing records of same city, to rectify it we can pass the index

In [7]:

```
df1 = pd.DataFrame({'city': ['delhi', 'mumbai', 'agra', 'goa'],\
                    'temperature': [20, 21, 19, 18]} , index=[0,1,2,3])
# 0,1,2,3 are the indexes given to 'delhi', 'mumbai', 'agra', 'goa'
df2 = pd.DataFrame({'city': ['agra','mumbai','goa','delhi'], \
                    'windspeed': [2, 2, 1, 1]}, index=[2, 1,3,0])
#index=[2, 1,3,0] are the indexes for 'agra','mumbai','goa','delhi'
```

In [115]:

```
df3 = pd.concat([df1,df2], axis =1)
df3
```

Out[115]:

	city	temperature	city	windspeed
0	delhi	20	delhi	1
1	mumbai	21	mumbai	2
2	agra	19	agra	2
3	goa	18	goa	1

Check what will happen if axis =0, it means rows

Append:

The concat method can combine data frames along either rows or columns, while the append method only combines data frames along rows

In [8]:

```
# dataframe 1
import pandas as pd

# assign data of lists.
data = {'city': ['delhi', 'mumbai', 'agra'], \
        'positive': [20, 21, 19],'neagtive': [120, 121, 119] }

# Create DataFrame
df1 = pd.DataFrame(data)
df1
```

Out[8]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119

In [117]:

```
# dataframe 2
import pandas as pd

# assign data of lists.
data = {'city': ['delhi', 'mumbai', 'agra'],\
        'positive': [210, 211, 19],'neagtive': [12, 121, 109] }

# Create DataFrame
df2 = pd.DataFrame(data)
df2
```

Out[117]:

	city	positive	neagtive
0	delhi	210	12
1	mumbai	211	121
2	agra	19	109

In [118]:

```
df3 = df1._append(df2)
```

In [119]:

```
df3
```

Out[119]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119
0	delhi	210	12
1	mumbai	211	121
2	agra	19	109

Merge data frames:In merging, you can merge two data frames to form a single data frame. You can also decide which columns you want to make common.

merge: always combine based on a column and we have to specify it, some column should be same in both dataframes based on which we can combine

In [120]:

```
df3 = df1.merge(df2, on = 'city')
df3
```

Out[120]:

	city	positive_x	neagtive_x	positive_y	neagtive_y
0	delhi	20	120	210	12
1	mumbai	21	121	211	121
2	agra	19	119	19	109

In [121]:

```
# positive_x    neagtive_x  belongs to first dataframe and positive_y
```

We can join the dataframes in different ways: 1)inner join: only common data of the dataframes are outputted 2)left join:That means we should get all records of left dataframe and only the matching data of right dataframe. 3)Right join:That means we should get all records of right dataframe and only the matching data of left dataframe. 4)Full outer join: all data from right and left dataframe. if no matching NaN will come

In [122]:

```
# inner join

df3 = df1.merge(df2, on = 'city', how ='inner')
df3
```

Out[122]:

	city	positive_x	neagtive_x	positive_y	neagtive_y
0	delhi	20	120	210	12
1	mumbai	21	121	211	121
2	agra	19	119	19	109

In [123]:

```
# in the above output we cant see the change as all records were common
```

In [124]:

```
# dataframe 1
import pandas as pd

# assign data of lists.
data = {'city': ['delhi', 'mumbai', 'agra', 'goa'], \
        'positive': [20, 21, 19, 88],'neagtive': [120, 121, 119, 133] }

# Create DataFrame
df1 = pd.DataFrame(data)
df1
```

Out[124]:

	city	positive	neagtive
0	delhi	20	120
1	mumbai	21	121
2	agra	19	119
3	goa	88	133

In [9]:

```
# dataframe 2
import pandas as pd

# assign data of lists.
data = {'city': ['delhi', 'mumbai', 'agra'], \
        'positive': [210, 211, 19],'neagtive': [12, 121, 109] }

# Create DataFrame
df2 = pd.DataFrame(data)
df2
```

Out[9]:

	city	positive	neagtive
0	delhi	210	12
1	mumbai	211	121
2	agra	19	109

In [126]:

```
df3 = df1.merge(df2, on = 'city', how ='inner')
df3
```

Out[126]:

	city	positive_x	neagtive_x	positive_y	neagtive_y
0	delhi	20	120	210	12
1	mumbai	21	121	211	121
2	agra	19	119	19	109

In [127]:

```
# we see that record for goa did not come as it was not common in both
```

In [128]:

```
# left join
df3 = df1.merge(df2, on = 'city', how ='left')
df3
```

Out[128]:

	city	positive_x	neagtive_x	positive_y	neagtive_y
0	delhi	20	120	210.0	12.0
1	mumbai	21	121	211.0	121.0
2	agra	19	119	19.0	109.0
3	goa	88	133	NaN	NaN

In [129]:

```
# Right join
df3 = df1.merge(df2, on = 'city', how ='right')
df3
```

Out[129]:

	city	positive_x	neagtive_x	positive_y	neagtive_y
0	delhi	20	120	210	12
1	mumbai	21	121	211	121
2	agra	19	119	19	109

In [130]:

```
# outer join
# Right join
df3 = df1.merge(df2, on = 'city', how ='outer')
df3
```

Out[130]:

	city	positive_x	neagtive_x	positive_y	neagtive_y
0	delhi	20	120	210.0	12.0
1	mumbai	21	121	211.0	121.0
2	agra	19	119	19.0	109.0
3	goa	88	133	NaN	NaN

https://github.com/codebasics/py/blob/master/pandas/9_merge/pandas_merge.ipynb (https://github.com/codebasics/py/blob/master/pandas/9_merge/pandas_merge.ipynb)