

Consider the following Python dictionary data and Python list labels:

```
data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',  
'spoonbills', 'spoonbills'], 'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4], 'visits': [2, 4, 3, 4, 3, 4,  
2, 2, 3, 2], 'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']}
```

```
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

1. Create a DataFrame birds from this dictionary data which has the index labels.

```
In [1]: import pandas as pd  
import numpy as np  
data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',  
'spoonbills', 'spoonbills'], 'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4],  
'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2], 'priority': ['yes',  
'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
  
birds = pd.DataFrame(data, index=labels)  
birds
```

Out[1]:

	age	birds	priority	visits
a	3.5	Cranes	yes	2
b	4.0	Cranes	yes	4
c	1.5	plovers	no	3
d	NaN	spoonbills	yes	4
e	6.0	spoonbills	no	3
f	3.0	Cranes	no	4

	age	birds	priority	visits
g	5.5	plovers	no	2
h	NaN	Cranes	yes	2
i	8.0	spoonbills	no	3
j	4.0	spoonbills	no	2

2. Display a summary of the basic information about birds DataFrame and its data.

In [2]: `birds.describe()`

Out[2]:

	age	visits
count	8.000000	10.000000
mean	4.437500	2.900000
std	2.007797	0.875595
min	1.500000	2.000000
25%	3.375000	2.000000
50%	4.000000	3.000000
75%	5.625000	3.750000
max	8.000000	4.000000

3. Print the first 2 rows of the birds dataframe

In [3]: `birds[0:2]`

Out[3]:

	age	birds	priority	visits
a	3.5	Cranes	yes	2

	age	birds	priority	visits
b	4.0	Cranes	yes	4

4. Print all the rows with only 'birds' and 'age' columns from the dataframe

In [4]: `birds[['birds', 'age']]`

Out[4]:

	birds	age
a	Cranes	3.5
b	Cranes	4.0
c	plovers	1.5
d	spoonbills	NaN
e	spoonbills	6.0
f	Cranes	3.0
g	plovers	5.5
h	Cranes	NaN
i	spoonbills	8.0
j	spoonbills	4.0

5. select [2, 3, 7] rows and in columns ['birds', 'age', 'visits']

In [5]: `print(birds['birds'].iloc[2], birds['age'].iloc[2],birds['visits'].iloc[2])
print(birds['birds'].iloc[3], birds['age'].iloc[3],birds['visits'].iloc[3])
print(birds['birds'].iloc[7], birds['age'].iloc[7],birds['visits'].iloc[7])`

plovers 1.5 3

```
spoonbills nan 4
Cranes nan 2
```

6. select the rows where the number of visits is less than 4

```
In [6]: birds[birds['visits']<4]
```

Out[6]:

	age	birds	priority	visits
a	3.5	Cranes	yes	2
c	1.5	plovers	no	3
e	6.0	spoonbills	no	3
g	5.5	plovers	no	2
h	NaN	Cranes	yes	2
i	8.0	spoonbills	no	3
j	4.0	spoonbills	no	2

7. select the rows with columns ['birds', 'visits'] where the age is missing i.e NaN

```
In [7]: birds[['birds', 'visits']][birds['age'].isnull()]
```

Out[7]:

	birds	visits
d	spoonbills	4
h	Cranes	2

8. Select the rows where the birds is a Cranes and the age is less than 4

```
In [8]: birds[['birds', 'age']][(birds['birds']=='Cranes') & (birds['age']<4)]
```

Out[8]:

	birds	age
a	Cranes	3.5
f	Cranes	3.0

9. Select the rows the age is between 2 and 4(inclusive)

```
In [9]: birds['age'][(birds['age']<=4) & (birds['age']>=2)]
```

```
Out[9]: a    3.5
        b    4.0
        f    3.0
        j    4.0
        Name: age, dtype: float64
```

10. Find the total number of visits of the bird Cranes

```
In [10]: birds[(birds['visits']>0) & (birds['birds']=='Cranes')].sum()    #summi
ng visits of Cranes
```

```
Out[10]: age                10.5
        birds    CranesCranesCranesCranes
        priority                yesyesnoyes
        visits                12
        dtype: object
```

11. Calculate the mean age for each different birds in dataframe.

```
In [11]: birds[['age']].mean()
```

```
Out[11]: age    4.4375
        dtype: float64
```

12. Append a new row 'k' to dataframe with your choice of values for each column. Then delete that row to return the original DataFrame.

```

In [12]: val = {'birds': ['Cranes'], 'age': [4],
               'visits': [2], 'priority': ['yes']}
          label = ['k']
          values = pd.DataFrame(val, index=label)

          birds=birds.append(values, ignore_index=True)

          birds

```

Out[12]:

	age	birds	priority	visits
0	3.5	Cranes	yes	2
1	4.0	Cranes	yes	4
2	1.5	plovers	no	3
3	NaN	spoonbills	yes	4
4	6.0	spoonbills	no	3
5	3.0	Cranes	no	4
6	5.5	plovers	no	2
7	NaN	Cranes	yes	2
8	8.0	spoonbills	no	3
9	4.0	spoonbills	no	2
10	4.0	Cranes	yes	2

```

In [13]: birds.drop([birds.index[10]])

```

Out[13]:

	age	birds	priority	visits
0	3.5	Cranes	yes	2
1	4.0	Cranes	yes	4

	age	birds	priority	visits
2	1.5	plovers	no	3
3	NaN	spoonbills	yes	4
4	6.0	spoonbills	no	3
5	3.0	Cranes	no	4
6	5.5	plovers	no	2
7	NaN	Cranes	yes	2
8	8.0	spoonbills	no	3
9	4.0	spoonbills	no	2

13. Find the number of each type of birds in dataframe (Counts)

In [13]: `birds.groupby(birds['birds']).count()`

Out[13]:

	age	birds	priority	visits
birds				
Cranes	4	5	5	5
plovers	2	2	2	2
spoonbills	3	4	4	4

14. Sort dataframe (birds) first by the values in the 'age' in descending order, then by the value in the 'visits' column in ascending order.

In [14]: `sorting = birds.sort(['age', 'visits'], ascending=[0,1])
print(sorting)`

```

      age      birds priority  visits
2  8.0  spoonbills      no        3
3  6.0  spoonbills      no        3
9  4.0  spoonbills      no        2

```

8	8.0	spoonbills	no	3
4	6.0	spoonbills	no	3
6	5.5	plovers	no	2
9	4.0	spoonbills	no	2
10	4.0	Cranes	yes	2
1	4.0	Cranes	yes	4
0	3.5	Cranes	yes	2
5	3.0	Cranes	no	4
2	1.5	plovers	no	3
7	NaN	Cranes	yes	2
3	NaN	spoonbills	yes	4

15. Replace the priority column values with 'yes' should be 1 and 'no' should be 0

In [15]: `birds.replace(['yes', 'no'], [1, 0])` *#x=priority*

Out[15]:

	age	birds	priority	visits
0	3.5	Cranes	1	2
1	4.0	Cranes	1	4
2	1.5	plovers	0	3
3	NaN	spoonbills	1	4
4	6.0	spoonbills	0	3
5	3.0	Cranes	0	4
6	5.5	plovers	0	2
7	NaN	Cranes	1	2
8	8.0	spoonbills	0	3
9	4.0	spoonbills	0	2
10	4.0	Cranes	1	2

16. In the 'birds' column, change the 'Cranes' entries to 'trumpeters'.


```
In [16]: birds.replace(['Cranes'], ['trumpeters'])  
ng index 10      # x = birds
```

Out[16]:

	age	birds	priority	visits
0	3.5	trumpeters	yes	2
1	4.0	trumpeters	yes	4
2	1.5	plovers	no	3
3	NaN	spoonbills	yes	4
4	6.0	spoonbills	no	3
5	3.0	trumpeters	no	4
6	5.5	plovers	no	2
7	NaN	trumpeters	yes	2
8	8.0	spoonbills	no	3
9	4.0	spoonbills	no	2
10	4.0	trumpeters	yes	2