

**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 [2]: 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']},  
        'no', 'no']}]  
  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
  
birds_df = pd.DataFrame(data, index = labels)  
print(birds_df)
```

	birds	age	visits	priority
a	Cranes	3.5	2	yes
b	Cranes	4.0	4	yes
c	plovers	1.5	3	no
d	spoonbills	NaN	4	yes
e	spoonbills	6.0	3	no
f	Cranes	3.0	4	no
g	plovers	5.5	2	no
h	Cranes	NaN	2	yes
i	spoonbills	8.0	3	no
j	spoonbills	4.0	2	no

```
j spoonbills 4.0 2 no
```

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

```
In [3]: 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', 'no', 'yes', 'no']}

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

birds_df = pd.DataFrame(data, index = labels)
print(birds_df.info())

<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
birds      10 non-null object
age         8 non-null float64
visits      10 non-null int64
priority    10 non-null object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None
```

## 3. Print the first 2 rows of the birds dataframe

```
In [4]: 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', 'no', 'yes', 'no']}
```

```
, 'no', 'no']}]

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

birds_df = pd.DataFrame(data , index = labels )
print(birds_df.head(2))
```

```
      birds  age  visits priority
a  Cranes  3.5      2      yes
b  Cranes  4.0      4      yes
```

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

```
In [5]: 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', 'no', 'yes', 'no', 'no']}

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

birds_df = pd.DataFrame(data , index = labels )
print(birds_df[['birds', 'age']])
```

```
      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 [6]: 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', 'no', 'yes', 'no', 'no']}

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

birds_df = pd.DataFrame(data, index = labels)
print(birds_df[['birds', 'age', 'visits']].iloc[[2,3,7]] )
```

	birds	age	visits
c	plovers	1.5	3
d	spoonbills	NaN	4
h	Cranes	NaN	2

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

```
In [7]: print(birds_df[(birds_df['visits'] < 4) ])
```

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

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

```
In [8]: print(birds_df[['birds', 'visits']][birds_df['age'].isnull()])
```

```

      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 [9]: print(birds_df[(birds_df['birds'] == 'Cranes') & (birds_df['age'] < 4)])
```

```

      birds  age  visits  priority
a  Cranes  3.5      2      yes
f  Cranes  3.0      4      no

```

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

```
In [10]: birds_df[ (birds_df['age'] < 4) & (birds_df['age'] >= 2)]
```

Out[10]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes
f	Cranes	3.0	4	no

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

```
In [11]: s = sum(birds_df['visits'])
print(s)
```

29

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

```
In [38]: birds_df.groupby("birds")["age"].mean()
```

Out[38]: birds

```
Cranes      3.5
plovers     3.5
spoonbills  6.0
Name: age, 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 [13]: birds_df.loc['k'] = ['owl' , 7 , 4 , 'yes']
print(birds_df)
print("*****")
birds_df =birds_df.drop('k')
print(birds_df)
```

```
      birds  age  visits  priority
a    Cranes  3.5      2      yes
b    Cranes  4.0      4      yes
c  plovers  1.5      3      no
d  spoonbills NaN      4      yes
e  spoonbills 6.0      3      no
f    Cranes  3.0      4      no
g  plovers  5.5      2      no
h    Cranes  NaN      2      yes
i  spoonbills 8.0      3      no
j  spoonbills 4.0      2      no
k      owl  7.0      4      yes
*****
      birds  age  visits  priority
a    Cranes  3.5      2      yes
b    Cranes  4.0      4      yes
c  plovers  1.5      3      no
d  spoonbills NaN      4      yes
e  spoonbills 6.0      3      no
f    Cranes  3.0      4      no
g  plovers  5.5      2      no
h    Cranes  NaN      2      yes
i  spoonbills 8.0      3      no
j  spoonbills 4.0      2      no
```

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

```
In [14]: birds_df.groupby("birds").count()
```

```
Out[14]:
```

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

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

```
In [15]: s1 =birds_df.sort_values(by = 'age' , ascending = False , na_position =  
'first')  
print(s1)  
print("*****")  
s2 =birds_df.sort_values(by = 'visits' , ascending = True )  
print(s2)
```

```
      birds  age  visits  priority  
d  spoonbills  NaN      4      yes  
h    Cranes  NaN      2      yes  
i  spoonbills  8.0      3      no  
e  spoonbills  6.0      3      no  
g    plovers  5.5      2      no  
b    Cranes  4.0      4      yes  
j  spoonbills  4.0      2      no  
a    Cranes  3.5      2      yes  
f    Cranes  3.0      4      no  
c    plovers  1.5      3      no  
*****  
      birds  age  visits  priority  
a    Cranes  3.5      2      ves
```

```

-         -         -         -         -
g    plovers  5.5      2      no
h    Cranes  NaN      2      yes
j    spoonbills  4.0    2      no
c    plovers  1.5      3      no
e    spoonbills  6.0    3      no
i    spoonbills  8.0    3      no
b    Cranes  4.0      4      yes
d    spoonbills  NaN    4      yes
f    Cranes  3.0      4      no

```

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

```
In [21]: birds_df.replace(to_replace = ['yes', 'no'], value = [1,0])
```

Out[21]:

	birds	age	visits	priority
<b>a</b>	trumpeters	3.5	2	1
<b>b</b>	trumpeters	4.0	4	1
<b>c</b>	plovers	1.5	3	0
<b>d</b>	spoonbills	NaN	4	1
<b>e</b>	spoonbills	6.0	3	0
<b>f</b>	trumpeters	3.0	4	0
<b>g</b>	plovers	5.5	2	0
<b>h</b>	trumpeters	NaN	2	1
<b>i</b>	spoonbills	8.0	3	0
<b>j</b>	spoonbills	4.0	2	0

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



In [23]: `birds_df.replace(to_replace="Cranes", value= "trumpeters")`

Out[23]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>a</b>	trumpeters	3.5	2	yes
<b>b</b>	trumpeters	4.0	4	yes
<b>c</b>	plovers	1.5	3	no
<b>d</b>	spoonbills	NaN	4	yes
<b>e</b>	spoonbills	6.0	3	no
<b>f</b>	trumpeters	3.0	4	no
<b>g</b>	plovers	5.5	2	no
<b>h</b>	trumpeters	NaN	2	yes
<b>i</b>	spoonbills	8.0	3	no
<b>j</b>	spoonbills	4.0	2	no