

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)
```

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

In [2]:

```
print(birds.columns)
print(birds.shape)
birds.describe()
```

```
Index(['birds', 'age', 'visits', 'priority'], dtype='object')
(10, 4)
```

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

In [3]:

```
birds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   birds       10 non-null     object
1   age         8 non-null      float64
2   visits      10 non-null     int64
3   priority    10 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

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

In [4]:

```
birds.head(2)
```

Out[4]:

	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]:

```
birds[['birds', 'age']]
```

Out[5]:

	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]:

```
birds[['birds', 'age', 'visits']].iloc[[1,2,6]]
```

Out[6]:

	<b>birds</b>	<b>age</b>	<b>visits</b>
<b>b</b>	Cranes	4.0	4
<b>c</b>	plovers	1.5	3
<b>g</b>	plovers	5.5	2

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

In [7]:

```
birds[birds.visits<4]
```

Out[7]:

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

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

In [8]:

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

Out[8]:

	<b>birds</b>	<b>visits</b>
<b>d</b>	spoonbills	4
<b>h</b>	Cranes	2

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

In [9]:

```
birds[birds['birds']=='Cranes'][birds.age<4]
```

C:\Softwares\Anaconda\lib\site-packages\ipykernel\_launcher.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
 """Entry point for launching an IPython kernel.

Out[9]:

	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[(birds.age>=2) & (birds.age<=4)]
```

Out[10]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes
b	Cranes	4.0	4	yes
f	Cranes	3.0	4	no
j	spoonbills	4.0	2	no

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

In [11]:

```
a=birds.groupby('birds')  
df=a.get_group('Cranes')  
print(df)  
df['visits'].sum()
```

	birds	age	visits	priority
a	Cranes	3.5	2	yes
b	Cranes	4.0	4	yes
f	Cranes	3.0	4	no
h	Cranes	NaN	2	yes

Out[11]:

12

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

In [12]:

```
a.mean()['age']
```

Out[12]:

```
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.loc['k']=[4, 'ab', 'no', 2]
print(birds)
birds=birds.drop('k',axis=0)
print(birds)
```

```
birds  age visits priority
a  Cranes  3.5      2     yes
b  Cranes   4      4     yes
c  plovers  1.5      3      no
d  spoonbills NaN     4     yes
e  spoonbills  6      3      no
f  Cranes   3      4      no
g  plovers  5.5      2      no
h  Cranes  NaN     2     yes
i  spoonbills  8      3      no
j  spoonbills  4      2      no
k      4  ab     no      2

birds  age visits priority
a  Cranes  3.5      2     yes
b  Cranes   4      4     yes
c  plovers  1.5      3      no
d  spoonbills NaN     4     yes
e  spoonbills  6      3      no
f  Cranes   3      4      no
g  plovers  5.5      2      no
h  Cranes  NaN     2     yes
i  spoonbills  8      3      no
j  spoonbills  4      2      no
```

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

In [14]:

```
print(birds.shape)
a=birds['birds'].value_counts()
print(a)
```

```
(10, 4)
Cranes      4
spoonbills  4
plovers     2
Name: birds, dtype: int64
```

**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]:

```
birds.sort_values(by=['age', 'visits'], ascending=[False, True])
```

Out[15]:

	birds	age	visits	priority
i	spoonbills	8	3	no
e	spoonbills	6	3	no
g	plovers	5.5	2	no
j	spoonbills	4	2	no
b	Cranes	4	4	yes
a	Cranes	3.5	2	yes
f	Cranes	3	4	no
c	plovers	1.5	3	no
h	Cranes	NaN	2	yes
d	spoonbills	NaN	4	yes

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

In [16]:

```
birds['priority'][birds.priority=='yes']=1  
birds['priority'][birds.priority=='no']=0  
print(birds)
```

	birds	age	visits	priority
a	Cranes	3.5	2	1
b	Cranes	4	4	1
c	plovers	1.5	3	0
d	spoonbills	NaN	4	1
e	spoonbills	6	3	0
f	Cranes	3	4	0
g	plovers	5.5	2	0
h	Cranes	NaN	2	1
i	spoonbills	8	3	0
j	spoonbills	4	2	0

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

In [17]:

```
birds['birds'][birds.birds=='Cranes']='trumpeters'
```

In [18]:

```
print(birds)
```

	birds	age	visits	priority
a	trumpeters	3.5	2	1
b	trumpeters	4	4	1
c	plovers	1.5	3	0
d	spoonbills	NaN	4	1
e	spoonbills	6	3	0
f	trumpeters	3	4	0
g	plovers	5.5	2	0
h	trumpeters	NaN	2	1
i	spoonbills	8	3	0
j	spoonbills	4	2	0

In [ ]: