

### 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', 'spo
onbills', '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', '
yes', 'no', 'no']}

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

birdsDF = pd.DataFrame({"birds":data['birds'],index=labels)
birdsDF
```

Out[2]:

	birds
a	Cranes
b	Cranes
c	plovers
d	spoonbills
e	spoonbills
f	Cranes
g	plovers
h	Cranes
i	spoonbills
j	spoonbills

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

```
In [36]: print(birdsDF.birds.describe())

#other way
print(birdsDF['birds'].describe())

count          14
unique          4
top      spoonbills
freq           4
Name: birds, dtype: object
count          14
unique          4
top      spoonbills
freq           4
Name: birds, dtype: object
```

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

```
In [37]: birdsDF.iloc[:2]
```

Out[37]:

	birds
a	Cranes
b	Cranes

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

```
In [57]: mp={}
#print(dir(df))
for item in data:
    mp[item]=data[item]

df=pd.DataFrame(mp)

#print all thr rows with only birds and age columns
df[['birds','age']]
```

Out[57]:

	<b>birds</b>	<b>age</b>
<b>0</b>	Cranes	3.5
<b>1</b>	Cranes	4.0
<b>2</b>	plovers	1.5
<b>3</b>	spoonbills	NaN
<b>4</b>	spoonbills	6.0
<b>5</b>	Cranes	3.0
<b>6</b>	plovers	5.5
<b>7</b>	Cranes	NaN
<b>8</b>	spoonbills	8.0
<b>9</b>	spoonbills	4.0

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

```
In [39]: df.loc[[2,3,7]][['birds','age','visits']]
```

Out[39]:

	<b>birds</b>	<b>age</b>	<b>visits</b>
<b>2</b>	plovers	1.5	3
<b>3</b>	spoonbills	NaN	4
<b>7</b>	Cranes	NaN	2

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

```
In [40]: df[df['visits']<4]
```

```
Out[40]:
```

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

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

```
In [41]: df[['birds','visits']][np.isnan(df.age)]
```

```
Out[41]:
```

	birds	visits
3	spoonbills	4
7	Cranes	2

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

```
In [42]: #df[df.birds=='Cranes'][df.age<4]  
  
df[(df["age"] < 4) & (df["birds"] == "Cranes")]
```

```
Out[42]:
```

	birds	age	visits	priority
0	Cranes	3.5	2	yes
5	Cranes	3.0	4	no

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

```
In [43]: df[(df.age>2) & (df.age<5)]
```

Out[43]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>0</b>	Cranes	3.5	2	yes
<b>1</b>	Cranes	4.0	4	yes
<b>5</b>	Cranes	3.0	4	no
<b>9</b>	spoonbills	4.0	2	no

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

```
In [44]: g=df.groupby(df.birds)
l=len(g.get_group('Cranes'))
print(l)
```

4

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

```
In [282]: g.mean()
```

Out[282]:

	<b>age</b>	<b>visits</b>
<b>birds</b>		
<b>Cranes</b>	3.5	3.0
<b>plovers</b>	3.5	2.5
<b>spoonbills</b>	6.0	3.0

## 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 [45]: newDF = pd.DataFrame({"birds":["Hen"]}, index=['k'])
birdsDF=birdsDF.append(newDF)
print(birdsDF)
```

```
      birds
a    Cranes
b    Cranes
c  plovers
d  spoonbills
e  spoonbills
f    Cranes
g  plovers
h    Cranes
i  spoonbills
j  spoonbills
k        Hen
k        Hen
k        Hen
k        Hen
k        Hen
```

```
In [46]: #delete that row to return the original DataFrame
birdsDF.drop(index='k')
```

Out[46]:

	<b>birds</b>
<b>a</b>	Cranes
<b>b</b>	Cranes
<b>c</b>	plovers
<b>d</b>	spoonbills
<b>e</b>	spoonbills
<b>f</b>	Cranes
<b>g</b>	plovers
<b>h</b>	Cranes
<b>i</b>	spoonbills
<b>j</b>	spoonbills

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

```
In [47]: ls=list(df['birds'])
mpBrd={};
for key in ls:
    if key in mpBrd:
        mpBrd[key] = mpBrd[key]+1
    else:
        mpBrd[key] = 1
print(mpBrd)

{'Cranes': 4, 'plovers': 2, 'spoonbills': 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 [98]: df.sort_values(['age','visits'],ascending=[False,True])
```

Out[98]:

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

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

```
In [94]: #df.replace('yes',1)
df=df.replace({'priority':{'No':0,'yes':1}})
df
```

Out[94]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>0</b>	trumpeter	3.5	2	1
<b>1</b>	trumpeter	4.0	4	1
<b>2</b>	plovers	1.5	3	0
<b>3</b>	spoonbills	NaN	4	1
<b>4</b>	spoonbills	6.0	3	0
<b>5</b>	trumpeter	3.0	4	0
<b>6</b>	plovers	5.5	2	0
<b>7</b>	trumpeter	NaN	2	1
<b>8</b>	spoonbills	8.0	3	0
<b>9</b>	spoonbills	4.0	2	0

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

```
In [70]: df=df.replace({'birds':{'Cranes':'trumpeter'}})
df
```

Out[70]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>0</b>	trumpeter	3.5	2	1
<b>1</b>	trumpeter	4.0	4	1
<b>2</b>	plovers	1.5	3	0
<b>3</b>	spoonbills	NaN	4	1
<b>4</b>	spoonbills	6.0	3	0
<b>5</b>	trumpeter	3.0	4	0
<b>6</b>	plovers	5.5	2	0
<b>7</b>	trumpeter	NaN	2	1
<b>8</b>	spoonbills	8.0	3	0
<b>9</b>	spoonbills	4.0	2	0