

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']
```

```
In [1]: import pandas as pd
import numpy as np
```

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

```
In [2]: 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']
```

```
In [3]: df = pd.DataFrame(data, index=labels)
df.head()
```

Out[3]:

	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

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

In [4]:

df

Out[4]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>a</b>	Cranes	3.5	2	yes
<b>b</b>	Cranes	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>	Cranes	3.0	4	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

In [5]: df.describe().transpose()

Out[5]:

	<b>count</b>	<b>mean</b>	<b>std</b>	<b>min</b>	<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>max</b>
<b>age</b>	8.0	4.4375	2.007797	1.5	3.375	4.0	5.625	8.0
<b>visits</b>	10.0	2.9000	0.875595	2.0	2.000	3.0	3.750	4.0

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

In [6]: df.head(2)

Out[6]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>a</b>	Cranes	3.5	2	yes
<b>b</b>	Cranes	4.0	4	yes

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

```
In [7]: df[['birds', 'age']]
```

Out[7]:

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

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

```
In [8]: df[['birds', 'age', 'visits']].iloc[[2, 3, 7]]
```

Out[8]:

	<b>birds</b>	<b>age</b>	<b>visits</b>
<b>c</b>	plovers	1.5	3
<b>d</b>	spoonbills	NaN	4
<b>h</b>	Cranes	NaN	2

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

```
In [9]: df[df.visits < 4]
```

```
Out[9]:
```

	<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 [10]: df[['birds', 'visits']][df.age.isnull()]
```

```
Out[10]:
```

	<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 [11]: df[(df.birds == 'Cranes') & (df.age < 4)]
```

```
Out[11]:
```

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>
<b>a</b>	Cranes	3.5	2	yes
<b>f</b>	Cranes	3.0	4	no

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

```
In [12]: df[(df.age >= 2) & (df.age <= 4)]
```

Out[12]:

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

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

```
In [13]: bird_groups = df.groupby('birds')
bird_groups.get_group('Cranes').visits.sum()
```

Out[13]: 12

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

```
In [14]: bird_groups.age.mean()
```

Out[14]: 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 [15]: k = pd.DataFrame([('Peacock', 5.5, 10, 'yes')], columns=['birds', 'age', 'visit', 'priority'])  
df = pd.concat([df, k], ignore_index=True)  
df
```

Out[15]:

	birds	age	visits	priority	visit
0	Cranes	3.5	2.0	yes	NaN
1	Cranes	4.0	4.0	yes	NaN
2	plovers	1.5	3.0	no	NaN
3	spoonbills	NaN	4.0	yes	NaN
4	spoonbills	6.0	3.0	no	NaN
5	Cranes	3.0	4.0	no	NaN
6	plovers	5.5	2.0	no	NaN
7	Cranes	NaN	2.0	yes	NaN
8	spoonbills	8.0	3.0	no	NaN
9	spoonbills	4.0	2.0	no	NaN
10	Peacock	5.5	NaN	yes	10.0

```
In [16]: df.drop([10], axis=0, inplace=True)
df
```

Out[16]:

	birds	age	visits	priority	visit
0	Cranes	3.5	2.0	yes	NaN
1	Cranes	4.0	4.0	yes	NaN
2	plovers	1.5	3.0	no	NaN
3	spoonbills	NaN	4.0	yes	NaN
4	spoonbills	6.0	3.0	no	NaN
5	Cranes	3.0	4.0	no	NaN
6	plovers	5.5	2.0	no	NaN
7	Cranes	NaN	2.0	yes	NaN
8	spoonbills	8.0	3.0	no	NaN
9	spoonbills	4.0	2.0	no	NaN

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

```
In [17]: bird_groups.birds.count()
```

```
Out[17]: birds
Cranes      4
plovers     2
spoonbills  4
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 [18]: df = df.sort_values(['age', 'visits'], ascending=[False, True])  
df
```

Out[18]:

	birds	age	visits	priority	visit
8	spoonbills	8.0	3.0	no	NaN
4	spoonbills	6.0	3.0	no	NaN
6	plovers	5.5	2.0	no	NaN
9	spoonbills	4.0	2.0	no	NaN
1	Cranes	4.0	4.0	yes	NaN
0	Cranes	3.5	2.0	yes	NaN
5	Cranes	3.0	4.0	no	NaN
2	plovers	1.5	3.0	no	NaN
7	Cranes	NaN	2.0	yes	NaN
3	spoonbills	NaN	4.0	yes	NaN

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



```
In [19]: df = df.replace({'priority': {'yes': 1, 'no': 0}})
df
```

Out[19]:

	birds	age	visits	priority	visit
8	spoonbills	8.0	3.0	0	NaN
4	spoonbills	6.0	3.0	0	NaN
6	plovers	5.5	2.0	0	NaN
9	spoonbills	4.0	2.0	0	NaN
1	Cranes	4.0	4.0	1	NaN
0	Cranes	3.5	2.0	1	NaN
5	Cranes	3.0	4.0	0	NaN
2	plovers	1.5	3.0	0	NaN
7	Cranes	NaN	2.0	1	NaN
3	spoonbills	NaN	4.0	1	NaN

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

```
In [20]: df = df.replace({'birds': {'Cranes': 'Trumpeters'}})
df
```

Out[20]:

	<b>birds</b>	<b>age</b>	<b>visits</b>	<b>priority</b>	<b>visit</b>
<b>8</b>	spoonbills	8.0	3.0	0	NaN
<b>4</b>	spoonbills	6.0	3.0	0	NaN
<b>6</b>	plovers	5.5	2.0	0	NaN
<b>9</b>	spoonbills	4.0	2.0	0	NaN
<b>1</b>	Trumpeters	4.0	4.0	1	NaN
<b>0</b>	Trumpeters	3.5	2.0	1	NaN
<b>5</b>	Trumpeters	3.0	4.0	0	NaN
<b>2</b>	plovers	1.5	3.0	0	NaN
<b>7</b>	Trumpeters	NaN	2.0	1	NaN
<b>3</b>	spoonbills	NaN	4.0	1	NaN

In [ ]: