

## 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 [94]: import pandas as pd #to create dataframe
import numpy as np #for dealing with 'nan' values

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', 'yes', 'no', 'no']}

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

df=pd.DataFrame(data,index=labels)

print(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
```

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

```
In [61]: print(df['birds'].describe())
```

```
count      10
unique       3
top      Cranes
freq         4
Name: birds, dtype: object
```

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

```
In [62]: print(df[0:2])
```

```
      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 [63]: print(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 [64]: print(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 [65]: print(df[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 [66]: print(df[df.age.isnull()][['birds', 'visits']])
```

	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 [67]: print(df[(df['age']<4) & (df['birds']=='Cranes')])
```

	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 [68]: print(df[(df['age']>=2) & (df['age']<=4)])
```

	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 [69]: df[df['birds']=='Cranes'].groupby(['birds'])['visits'].agg(sum)
```

```
Out[69]: birds
Cranes    12
Name: visits, dtype: int64
```

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

```
In [70]: df.groupby(['birds'])['age'].mean()
```

```
Out[70]: 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 [71]: row = {'birds': ['Cranes'], 'age': [4], 'visits': [2], 'priority': ['ye
```

```
s']}]}
```

```
df=df.append(pd.DataFrame(row,index=['k']))  
df.drop(['k'])
```

Out[71]:

	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 [72]: print(df.groupby(['birds']).size())
```

```
birds  
Cranes      5  
plovers     2  
spoonbills  4  
dtype: int64
```

### 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 [95]: print(df.sort_values(['age','visits'],ascending=[False,True]))
```

	birds	age	visits	priority
i	spoonbills	8.0	3	no
e	spoonbills	6.0	3	no
g	plovers	5.5	2	no
j	spoonbills	4.0	2	no
b	Cranes	4.0	4	yes
a	Cranes	3.5	2	yes
f	Cranes	3.0	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 [100]: df=df.replace({'priority':{'yes':'1','no':'0'}})
          print(df)
```

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

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

```
In [101]: df=df.replace({'birds':{'Cranes':'trumpeters'}})
          print(df)
```

	birds	age	visits	priority
a	trumpeters	3.5	2	1
b	trumpeters	4.0	4	1

```
c    plovers 1.5    3    0
d  spoonbills NaN    4    1
e  spoonbills 6.0    3    0
f  trumpeters 3.0    4    0
g    plovers 5.5    2    0
h  trumpeters NaN    2    1
i  spoonbills 8.0    3    0
j  spoonbills 4.0    2    0
```

In [ ]: