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 [53]:

```
import pandas as pd
import numpy as np
birds = pd.DataFrame({'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoon
        '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'
birds
```

Out[53]:

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

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

In [54]:

```
print(birds.describe())
```

```
            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
```

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

In [55]:

```
print(birds.iloc[:2])
```

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

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

In [56]:

```
print(birds[['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 [57]:

```
new = birds[['birds','age','visits']].iloc[[1,2,6]]
print(new)
```

```
     birds  age  visits
b   Cranes  4.0       4
c  plovers  1.5       3
g  plovers  5.5       2
```

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

In [58]:

```
n = birds[birds['visits']<4]
print(n)
```

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

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

In [59]:

```
mis = birds[birds.age.isnull()][['birds','visits']]
print(mis) # can't figure out why it's not showing any row values as 2 are with age
```

```
        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 [60]:

```
bir = birds[birds['birds']=="Cranes"][birds['age']<4]
print(bir)
```

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

/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:1: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

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

In [61]:

```
br = birds[birds['age']>=2][birds['age']<=4]
print(br)
```

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

/usr/local/lib/python3.5/dist-packages/ipykernel_launcher.py:1: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

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

In [62]:

```
su = birds['visits'][birds['birds']=='Cranes'].sum()
print(su)
```

```
12
```

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

In [63]:

```
b = birds.birds.unique()
l = []
for i in b:
    l.append((birds['age'][birds['birds']==i]).mean())
print(l)
```

```
[3.5, 3.5, 6.0]
```

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 [ ]:

```

```

In [64]:

```
k = pd.DataFrame({'birds':['crow'],'age':[3],'visits':[3],'priority':['yes']},index
birds = pd.concat([birds,k])
print(birds)
```

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

In [65]:

```
birds.drop(birds.index[-1],inplace = True)
print(birds)
```

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

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

In [66]:

```

count = birds['birds'].value_counts()
print(count)
```

```
spoonbills    4
Cranes        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 [67]:

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

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

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

In [68]:

```
birds = birds.replace('yes',1)
birds = birds.replace('no',0)
print(birds)
```

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

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

In [69]:

```
birds = birds.replace('Cranes','trumpeters')
print(birds)
```

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

In [ ]: