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]:
```
#Importing Pandas library
import pandas as pd
import numpy as np
#dictionary data
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']
#Creatng a DataFrame Birds and naming it df
df = pd.DataFrame(data,index=labels)

#Converting age column to numeric as there is a string value which had to conv
erted to missing value NAN
df['age']=pd.to_numeric(df['age'],errors='coerce')
print("DataFrame birds is")
print(df)
```

```
DataFrame birds is
        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 [126]:
```
df.describe()
```

Out[126]:

	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 [127]:
```
df.head(2)
```

Out[127]:

	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 [128]:
```
df[['birds','age']]
```

Out[128]:

	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 [129]:
```
#Selecting using iloc function which uses Indices
print(df[['birds','age', 'visits']].loc['b'])
print('**********************************')
print(df[['birds','age', 'visits']].loc['c'])
print('**********************************')
print(df[['birds','age', 'visits']].loc['g'])
```

```
birds     Cranes
age            4
visits         4
Name: b, dtype: object
**********************************
birds     plovers
age           1.5
visits          3
Name: c, dtype: object
**********************************
birds     plovers
age           5.5
visits          2
Name: g, dtype: object
```

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

In [130]:
```
df[df.visits<4]
```

Out[130]:

	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 [131]:
```
df_new=df[df.age.isnull()==True]
df_new[['birds', 'visits']]
```

Out[131]:

	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 [132]:
```
df[(df['birds'] == 'Cranes') & (df['age']< 4)]
```

Out[132]:

	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 [133]:
```
df[(df['age'] >= 2) & (df['age'] <= 4)]
```

Out[133]:

	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 [134]:
```
g= df.groupby('birds')
g.sum()
```

Out[134]:

	age	visits
birds		
Cranes	10.5	12
plovers	7.0	5
spoonbills	18.0	12

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

In [135]:
```
g.mean()
```

Out[135]:

birds	age	visits
Cranes	3.5	3.0
plovers	3.5	2.5
spoonbills	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 [136]:
```
print("DataFrame birds is:")
print(df)
print('*****************************************')
print("New DataFrame birds is:")
#Using loc function to append new row into DF
df.loc['k']=['ABC',5,55,'yes']
print(df)
print('*****************************************')
#Dropping the new row using drop function
df = df.drop('k')
print(df)
```

```
DataFrame birds is:
        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
*****************************************
New DataFrame birds is:
        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
k         ABC  5.0      55      yes
*****************************************
        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 [137]:
```
df['birds'].describe()
```

Out[137]:
```
count            10
unique            3
top       spoonbills
freq              4
Name: birds, dtype: object
```

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 [3]:
```
df.sort_values(by=['age','visits'], inplace=True,ascending=[False,True])
print (df)
```

```
        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 [138]:
```
df["priority"].replace({"yes": 1, "no": 0}, inplace=True)
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 [140]:
```
df['birds'].replace({"Cranes": 'trumpeters'}, inplace=True)
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 [ ]:
```
```