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 [27]:
```
import pandas as pd
import numpy as np

data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes', 'spoonbills', 'spoonbills'], 'age': [3.5, 4, 1.5,

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

#create a dataframe first
dataframe = pd.DataFrame(data,index=labels)

print(dataframe)
```

```
        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 [9]:
```
print(dataframe.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 [10]:
```
print(dataframe.head(2))
```

```
    birds  age  visits priority
0  Cranes  3.5       2      yes
1  Cranes  4.0       4      yes
```

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

In [28]:
```
#print(dataframe['birds'])
#print(dataframe['age'])
#dataframe = pd.DataFrame(data,index=False)
print(dataframe[dataframe.columns[0:3]])
```

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

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

In [38]:
```
print(dataframe['birds'].iloc[2], "   ",dataframe['age'].iloc[2],"  ",dataframe['visits'].iloc[2])
print(dataframe['birds'].iloc[3], "",dataframe['age'].iloc[3],"  ",dataframe['visits'].iloc[3])
print(dataframe['birds'].iloc[7],"    ", dataframe['age'].iloc[7],"  ",dataframe['visits'].iloc[3])
```

```
plovers     1.5    3
spoonbills  nan    4
Cranes      nan    4
```

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

In [39]:
```
print(dataframe[dataframe['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 [40]:
```
print(dataframe[dataframe['age'].isnull()])
```

```
        birds  age  visits priority
d  spoonbills  NaN       4      yes
h      Cranes  NaN       2      yes
```

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

In [56]:
```
#dataframe[dataframe.columns[0:3]]
sumValue = dataframe[(dataframe['birds'] == 'Cranes') & (dataframe['age'] < 4)]
print(sumValue)
```

```
    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 [53]:
```
print(dataframe[(dataframe['age'] >= 2) & (dataframe['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 [98]:
```
#print(dataframe[(dataframe['birds'] == 'Cranes') & (dataframe['visits'] > 0)].sum())
#print(dataframe.groupby(dataframe['birds']=='Cranes').sum()& dataframe['visits'])
df= pd.DataFrame(dataframe[(dataframe['birds'] == 'Cranes') & (dataframe['visits'] > 0)].sum())
print(df.iloc[2])
```

```
0    12
Name: visits, dtype: object
```

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

In [99]:
```
print(dataframe[['age']].mean())
```

```
age    4.4375
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 [113...
```
print("Append New row k to dataframe.")
dataframe.loc['k'] = ['peacock', 12.4, 5,'yes']
print(dataframe)
delete = dataframe.drop('k')
print()
print('Delete the row return original dataframe')
print(delete)

#dataframe.append(s, ignore_index=True)
#dataframe.drop([df.index[9]])
```

```
Append New row k to dataframe.
        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     peacock  12.4       5      yes

Delete the row return original dataframe
        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 [122...
```
# counting the duplicates
dups = dataframe.pivot_table(index = ["birds"], aggfunc ='size')
print(dups)
```

```
birds
Cranes         4
peacock        1
plovers        2
spoonbills     4
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 [128...
```
#sort_by_age = dataframe.sort_values('age')
sort_by_age = dataframe.sort_values(by='age', ascending=True)
print('Decending order of age')
print(sort_by_age)

print('Ascending order of age')
sort_by_age = dataframe.sort_values(by='visits', ascending=False)
print(sort_by_age)
```

```
Decending order of age
        birds   age  visits priority
c     plovers  1.5       3       no
f      Cranes  3.0       4       no
a      Cranes  3.5       2      yes
b      Cranes  4.0       4      yes
j  spoonbills  4.0       2       no
g     plovers  5.5       2       no
e  spoonbills  6.0       3       no
i  spoonbills  8.0       3       no
k     peacock  12.4       5      yes
d  spoonbills  NaN       4      yes
h      Cranes  NaN       2      yes
Ascending order of age
        birds   age  visits priority
k     peacock  12.4       5      yes
b      Cranes  4.0       4      yes
d  spoonbills  NaN       4      yes
f      Cranes  3.0       4       no
c     plovers  1.5       3       no
e  spoonbills  6.0       3       no
i  spoonbills  8.0       3       no
a      Cranes  3.5       2      yes
g     plovers  5.5       2       no
h      Cranes  NaN       2      yes
j  spoonbills  4.0       2       no
```

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

In [131...
```
#dataframe.priority.map(dict(yes=1, no=0))
dataframe['priority'] = dataframe['priority'].map({'yes': 1, 'no': 0})
print(dataframe)
```

```
        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
k     peacock  12.4       5         1
```

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

In [134...
```
print(dataframe.birds.map(lambda x: 'trumpeters' if x=='Cranes' else x))
```

```
a    trumpeters
b    trumpeters
c       plovers
d    spoonbills
e    spoonbills
f    trumpeters
g       plovers
h    trumpeters
i    spoonbills
j    spoonbills
k      peacock
Name: birds, dtype: object
```

In [ ]:
```

```