In [2]:

```
import pandas as pd
import numpy as np
```

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 [3]:

```
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 [4]:

```
pd.DataFrame.from_dict(data,orient = 'index' , columns = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'] )
```

Out[4]:

	a	b	c	d	e	f	g	h	i
birds	Cranes	Cranes	plovers	spoonbills	spoonbills	Cranes	plovers	Cranes	spoonbills
age	3.5	4	1.5	NaN	6	3	5.5	NaN	8
visits	2	4	3	4	3	4	2	2	3
priority	yes	yes	no	yes	no	no	no	yes	no

In [5]:

```
pd.DataFrame.from_dict(data,orient = 'index' , columns = labels)
```

Out[5]:

	a	b	c	d	e	f	g	h	i
birds	Cranes	Cranes	plovers	spoonbills	spoonbills	Cranes	plovers	Cranes	spoonbills
age	3.5	4	1.5	NaN	6	3	5.5	NaN	8
visits	2	4	3	4	3	4	2	2	3
priority	yes	yes	no	yes	no	no	no	yes	no

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

Summary

In the dataset we have a total of 3 unique birds.The features we have are name,age,number of visits and the priority of the birds.There are some birds whose age we dont know hence we have not defined data in their respective column.The visit column consist of numberic data while the priority column consist of boolean data.

3. Print the first 2 rows of the birds dataframe

In [6]:

```
df = pd.DataFrame.from_dict(data,orient = 'columns' )
#print(df)
print(df.loc[0:1])
```

```
    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 [7]:

```
print(df.loc[: ,['birds' , 'age']])
```

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

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

In [8]:

```
print(df.loc[[2,3,7] ,['birds' , 'age' , 'visits']])
```

```
         birds  age  visits
2     plovers  1.5       3
3  spoonbills  NaN       4
7       Cranes  NaN       2
```

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

In [9]:

```
print(df.loc[df['visits'] < 4 ])
```

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

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

In [10]:

```
print(df.loc[df['age'].isnull() , ['birds' , 'visits']])
```

```
        birds  visits
3  spoonbills       4
7      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]:

	birds	age	visits	priority
0	Cranes	3.5	2	yes
5	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
0	Cranes	3.5	2	yes
5	Cranes	3.0	4	no

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

In [13]:

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

Out[13]:

```
12
```

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

In [41]:

```
## df.groupby('birds')['age'].mean()

df_2 = df.groupby('birds')

df_2.mean()['age']
```

Out[41]:

```
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 [67]:

```
df_4 = df.append([{'birds' : 'plovers','age' : 3 , 'visits' : 4 , 'priority' : 'Yes'}]
, ignore_index = True)

df_4

# df_4 is our data frame with a new row added up.
```

Out[67]:

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

In [68]:

```
df_4.drop(10, inplace = True)

print(df_4)
```

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

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

In [43]:

```
df_3 = df.groupby('birds')
df_3['birds'].count()
```

Out[43]:

```
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 [69]:

```
df.sort_values(by = 'age' , ascending = False)
```

Out[69]:

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

In [70]:

```
df.sort_values(by = 'visits')
```

Out[70]:

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

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

In [85]:

```
df['priority'].replace(to_replace = ['yes' , 'no'],value = ['1','0'], inplace = True)
df
```

Out[85]:

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

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

In [87]:

```
df['birds'].replace(to_replace = 'Cranes' , value = 'trumpeters' , inplace = True)
df
```

Out[87]:

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

In [ ]:

```

```