In [137]:
```
import pandas as pd
import numpy as np
```

In [138]:
```
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 [139]:
```
df = pd.DataFrame.from_dict(data)#importing data from dictionary
df
```

Out[139]:

	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

In [140]:
```
df = pd.DataFrame(labels)#importing labels data in dataframe
df
```

Out[140]:

	0
0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j

In [141]:
```
df = pd.DataFrame(data,index = labels)#dataframe with index as labels
df
```

Out[141]:

	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 [142]:
```
print(df.info(data)) #printing information of data
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
birds       10 non-null object
age         8 non-null float64
visits      10 non-null int64
priority    10 non-null object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None
```

3. Print the first 2 rows of the birds dataframe

In [143]:
```
df[0:2] #slicing of data frame to print first two rows
```

Out[143]:

	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 [144]:
```
df[["birds","age"]] #printing particular columns
```

Out[144]:

	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 [145]:
```
d = df[['birds','age','visits']]#assigning value to the 'd'
d.iloc[[2,3,7]] #selecting particular rows i.e 2 = c,3 = d,h = 7
```

Out[145]:

	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 [146]:
```
df[df["visits"] < 4] #selecting rows in which the visits is less than 4
```

Out[146]:

	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 [147]:
```
df[df["age"].isnull()]
```

Out[147]:

	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 [165]:
```
df[(df["birds"] == "cranes") & (df["age"]<4)]
```

Out[165]:

	birds	age	visits	priority

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

In [149]:
```
df[(df["age"] >=2) & (df["age"]<=4)]
```

Out[149]:

	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 [171]:
```
df[(df["birds"] == "cranes") & (df["visits"])]
```

Out[171]:
```
birds       0.0
age         0.0
visits      0.0
priority    0.0
dtype: float64
```

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

In [172]:
```
df[["age"]].mean()
```

Out[172]:
```
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 [174]:
```
new_row = pd.DataFrame({"birds":"abc","age":3,"visits":2,"priority":"yes"})
new_row
```

Out[174]:

	age	birds	priority	visits
0	3	abc	yes	2

In [182]:
```
df.append(new_row,ignore_index = True,sort = True)
df.drop([df.index[1]])
```

Out[182]:

	birds	age	visits	priority
a	Cranes	3.5	2	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 [183]:
```
df.groupby(df["birds"]).count()
```

Out[183]:

	age	visits	priority
birds			
Cranes	3	4	4
plovers	2	2	2
spoonbills	3	4	4

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 [192]:
```
sort_age = df.sort_values("age",ascending = True)
print(sort_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
d  spoonbills  NaN       4      yes
h      Cranes  NaN       2      yes
```

In [193]:
```
sort_visits = df.sort_values("visits",ascending = False)
print(sort_visits)
```

```
          birds  age  visits priority
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 [206]:
```
df.priority.map(dict(yes = 1 ,no = 0))
```

Out[206]:
```
a    1
b    1
c    0
d    1
e    0
f    0
g    0
h    1
i    0
j    0
Name: priority, dtype: int64
```

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

In [221]:
```
df.birds.map(lambda x: "trumpeters" if x == "Cranes" else x)
```

Out[221]:
```
a    trumpeters
```