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 [1]:

```
import pandas as pd
import numpy as np
bird_data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plover
s', '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']}
df=pd.DataFrame(bird_data,index=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'])
df
```

Out[1]:

	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 [2]:

```
df.describe()
```

Out[2]:

	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 [3]:

```
df[:2]
```

Out[3]:

	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 [4]:

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

Out[4]:

	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 [5]:

```
df.iloc[[2,3,7],[1,0,3]]
```

Out[5]:

	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 [6]:

```
df[df.visits<4]
```

Out[6]:

	age	birds	priority	visits

	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 [7]:

```
p=df[df.age.isnull()]
p.iloc[:,[1,3]]
```

Out[7]:

	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 [8]:

```
q=df[df.birds=='Cranes']
q[q.age<4]
```

Out[8]:

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

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

In [9]:

```
df.loc[(df['age'] >= 2) & (df['age'] <= 4)]
```

Out[9]:

	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

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

In [10]:

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

Out[10]:

12

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

In [11]:

```
b=df.groupby('birds')
b

for birds,bird in b:
    print(birds)
    print(bird.age.mean())
    print("********")
```

```
Cranes
3.5
********
plovers
3.5
********
spoonbills
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 [12]:

```
df = df.append({'age' : '6' , 'birds' : 'Parrot','priority':'yes','visits':'7'} , ignore_index=True
)
df
```

Out[12]:

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

In [13]:

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

Out[13]:

	age	birds	priority	visits
0	3.5	Cranes	yes	2

	age	birds	priority	visits
1	4	Cranes	yes	4
2	1.5	plovers	no	3
3	NaN	spoonbills	yes	4
4	6	spoonbills	no	3
5	3	Cranes	no	4
6	5.5	plovers	no	2
7	NaN	Cranes	yes	2
8	8	spoonbills	no	3
9	4	spoonbills	no	2

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

In [14]:

```
g=df.groupby('birds').size()
print(g)
```

```
birds
Cranes         4
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 [15]:

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

Out[15]:

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

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

In [16]:

```
df.replace('yes',1,inplace=True)
df.replace('no',0,inplace=True)
df
```

Out[16]:

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

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

In [17]:

```
df['birds']=df.birds.apply(lambda y: y.replace('Cranes','trumpeters'))
df
```

Out[17]:

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