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', 'no', '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 [0]:
```
import pandas as pd
from pandas import DataFrame

import numpy as np
df= pd.DataFrame({'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', 'no', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']},index= ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'])
#df = DataFrame (data, columns = ['Birds','Age','Visits','Priority'],index=[1,2,3,4,5,6,7])
df
```

Out[0]:

	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 [0]:
```
df.info()
```

```
<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
```

3. Print the first 2 rows of the birds dataframe

In [0]:
```
df[:2]
```

Out[0]:

	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 [0]:
```
d=df[['Birds','Visits']]
d
```

Out[0]:

	Birds	Visits
a	Cranes	2
b	Cranes	4
c	plovers	3
d	spoonbills	4
e	spoonbills	3
f	Cranes	4
g	plovers	2
h	Cranes	2
i	spoonbills	3
j	spoonbills	2

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

In [0]:
```
#Based on Index starts from 0
df=df[['Birds','Age','Visits']]
df.iloc[[2, 3, 7], :]
```

Out[0]:

	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 [0]:
```
d2=df[df['Visits'] < 4]
d2
```

Out[0]:

	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 [0]:
```
df=df[df.isnull().any(axis=1)]
df[['Birds','Visits']]
```

Out[0]:

	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 [0]:
```
# Create variable with TRUE if Birds is Equal to 'Cranes'
s1= df['Birds'] == "Cranes"

# Create variable with TRUE if age is less than 4
s2 = df['Age'] <4


df[s1 & s2]
```

Out[0]:

	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 [0]:
```
df[(df['Age'] >= 2) & (df['Age'] <= 4)]
```

Out[0]:

	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 [0]:
```
v1=df[df['Birds']== "Cranes"]
#print("The total number of visits:",v1.sum().sum())
print("The Total Number of vists of the Cranes bird cane::",v1['Visits'].sum())
```

```
The Total Number of vists of the Cranes bird cane:: 12
```

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

In [0]:
```
m1=df['Age']
print("The mean age of birds::",m1.mean())
```

```
The mean age of birds:: 4.4375
```

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 [0]:
```
#Appending New Row
r1 = pd.DataFrame({'Birds':['parraot'],'Age':[4],'Visits':[5],'Priority':['yes']},index=['k'])
df=df.append(r1)
df
```

Out[0]:

	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	parraot	4.0	5	yes

In [0]:
```
#Deleting added row

df=df.drop('k')
df
```

Out[0]:

	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 [0]:
```
df.groupby('Birds').size()
```

Out[0]:
```
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 [0]:
```
df=df.sort_values('Age',ascending=False)
```

Out[0]:

	Birds	Age	Visits	Priority
i	spoonbills	8.0	3	no
e	spoonbills	6.0	3	no
g	plovers	5.5	2	no
b	Cranes	4.0	4	yes
j	spoonbills	4.0	2	no
a	Cranes	3.5	2	yes
f	Cranes	3.0	4	no
c	plovers	1.5	3	no
d	spoonbills	NaN	4	yes
h	Cranes	NaN	2	yes

In [0]:
```
# sorting by Visits(ascending order)
df.sort_values('Visits',ascending=True)
```

Out[0]:

	Birds	Age	Visits	Priority
a	Cranes	3.5	2	yes
g	plovers	5.5	2	no
h	Cranes	NaN	2	yes
j	spoonbills	4.0	2	no
c	plovers	1.5	3	no
e	spoonbills	6.0	3	no
i	spoonbills	8.0	3	no
b	Cranes	4.0	4	yes
d	spoonbills	NaN	4	yes
f	Cranes	3.0	4	no

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

In [0]:
```
df.Priority[data.Priority == 'yes'] = 1
df.Priority[data.Priority== 'no'] = 0
df
```

Out[0]:

	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 [0]:
```
#re=df
#re=re['Birds'].replace('Cranes', 'trumpeters')
df.Birds[data.Birds == 'Cranes'] = 'trumpeters'
df
```

Out[0]:

	Birds	Age	Visits	Priority
a	trumpeters	3.5	2	yes
b	trumpeters	4.0	4	yes
c	plovers	1.5	3	no
d	spoonbills	NaN	4	yes
e	spoonbills	6.0	3	no
f	trumpeters	3.0	4	no
g	plovers	5.5	2	no