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 [0]:

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

In [0]:

```
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 [0]:

```

```

1. Create a DataFrame birds from this dictionary data which has the index labels.

In [25]:

```
df1=pd.DataFrame(data,index=labels)
df1
```

Out[25]:

	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

In [0]:

```

```

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

In [23]:

```
df14['birds'].describe()
```

Out[23]:

```
count             10
unique             3
```

```
unique              3
top        spoonbills
freq                4
Name: birds, dtype: object
```

In [0]:

```
```

3. Print the first 2 rows of the birds dataframe

In [22]:

```
df14['birds'][:2]
```

Out[22]:

```
a     Cranes
b     Cranes
Name: birds, dtype: object
```

In [0]:

```
```

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

In [21]:

```
df14[['birds','age']]
```

Out[21]:

	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**

In [0]:

```
```

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

In [0]:

```
df14=pd.DataFrame(data,index=labels)
```

In [0]:

```
df14.loc[df14.loc['birds']]
```

In [20]:

```
df14[['birds','age', 'visits']].iloc[[2,3,7]]
```

Out[20]:

	birds	age	visits
c	plovers	1.5	3
d	spoonbills	NaN	4
h	Cranes	NaN	2

In [0]:

```
```

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

In [0]:

```
df12=pd.DataFrame(data,index=labels)
```

In [0]:

```
df12[df12['visits'] < 4]
```

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

In [0]:

```
```

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

In [0]:

```
df11=pd.DataFrame(data,index=labels)
```

In [0]:

```
t1=df11[df11['age'].isnull()]
t1
```

Out[0]:

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

In [0]:

```
t1.drop(columns=['age','priority'])
```

Out[0]:

birds	visits

	birds	visits
d	spoonbills	4
h	Cranes	2

In [0]:

```
t1.columns.values
```

Out[0]:

```
array(['birds', 'age', 'visits', 'priority'], dtype=object)
```

In [0]:

```

```

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

In [0]:

```
df10=pd.DataFrame(data,index=labels)
df10
```

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

In [0]:

```
df10[(df10['birds']=='Cranes') & (df10['age']<4)]
```

Out[0]:

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

In [0]:

```

```

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

In [0]:

```
df3=pd.DataFrame(data,index=labels)
df3
```

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

In [0]:

```
## https://stackoverflow.com/questions/31617845/how-to-select-rows-in-a-dataframe-between-two-values-in-python-pandas/40442778
```

In [0]:

```
df3 = df3[(df3['age'] >=2) & (df3['age'] <= 4)]
df3
```

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

In [0]:

```

```

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

In [0]:

```
df43=pd.DataFrame(data,index=labels)
df43
```

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

In [0]:

```
t1=df43.groupby([df43['birds']=='Cranes'])['visits'].sum()
t1
```

Out[0]:

```
birds
False    17
True     12
Name: visits, dtype: int64
```

There are 12 visits of bird cranes

In [0]:

```

```

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

In [0]:

```
df9=pd.DataFrame(data,index=labels)
df9
```

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

In [0]:

```
df9.groupby( 'birds' , as_index=False)['age'].mean()
```

Out[0]:

	birds	age
0	Cranes	3.5
1	plovers	3.5
2	spoonbills	6.0

In [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 [0]:

```
df6=pd.DataFrame(data,index=labels)
```

```
df6
```

In [0]:

```
#f7 = pd.DataFrame([k, k,k,k],columns=['birds','age','visits','priority'])
dict1={'birds':67,
            'age': 65,
            'visits': 86,
            'priority': 76}
df6.append(dict1,ignore_index=True)
```

Out[0]:

	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	67	65.0	86	76

In [0]:

```
#https://stackoverflow.com/questions/26921651/how-to-delete-the-last-row-of-data-of-a-pandas-dataframe
```

In [0]:

```
# Delete these row indexes from dataFrame
df6.drop(df6.tail(1).index,inplace=True)
```

In [0]:

```
df6
```

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

In [0]:

```

```

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

In [0]:

```
df5=pd.DataFrame(data,index=labels)
df5
```

In [0]:

```
df5['birds'].value_counts()
```

Out[0]:

```
spoonbills    4
Cranes        4
plovers       2
Name: birds, dtype: int64
```

In [0]:

```

```

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]:

```
df3=pd.DataFrame(data,index=labels)
df3
```

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

In [0]:

```
df3.sort_values(by=['age'],ascending=False,inplace=True)
df3
```

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

	birds	age	visits	priority
c	plovers	1.5	3	no
d	spoonbills	NaN	4	yes
h	Cranes	NaN	2	yes

In [0]:

```
df3.sort_values(by=['visits'], axis=0, ascending=True,inplace=True)
df3
```

Out[0]:

	birds	age	visits	priority
g	plovers	5.5	2	no
j	spoonbills	4.0	2	no
a	Cranes	3.5	2	yes
h	Cranes	NaN	2	yes
i	spoonbills	8.0	3	no
e	spoonbills	6.0	3	no
c	plovers	1.5	3	no
b	Cranes	4.0	4	yes
f	Cranes	3.0	4	no
d	spoonbills	NaN	4	yes

In [0]:

```
df4=pd.DataFrame(data,index=labels)
df4
```

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

In [0]:

```

```

In [0]:

```
df4.sort_values(by=['visits'],axis=0)
df4
```

Out[0]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes

	birds	age	visits	priority
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

In [0]:

```
```

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

In [0]:

```
dfq=pd.DataFrame(data,index=labels)
```

In [0]:

```
replace_values = {'yes' : 1, 'no' : 0 }


dfq= dfq.replace({"priority": replace_values})
dfq
```

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

In [0]:

```
```

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

In [0]:

```
df1=pd.DataFrame(data,index=labels)
df1
```

Out[0]:

	birds	age	visits	priority

	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

In [0]:

```
df1.replace('Cranes','trumpeters')
```

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
h	trumpeters	NaN	2	yes
i	spoonbills	8.0	3	no
j	spoonbills	4.0	2	no

In [0]:

```

```

In [0]:

```

```