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 [29]:

```
import numpy as np
import pandas as pd
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']

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

Out[29]:

	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 [30]:

```
df.describe()
```

Out[30]:

	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 [31]:

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

Out[31]:

	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 [32]:

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

Out[32]:

	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 [33]:

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

Out[33]:

	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 [34]:

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

Out[34]:

	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 [35]:

```
df[df['age'].isnull()]
```

Out[35]:

	birds	age	visits	priority
3	spoonbills	NaN	4	yes
7	Cranes	NaN	2	yes

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

In [36]:

```
df.loc[(df['birds'] == 'Cranes') & (df['age'] <4)]
```

Out[36]:

	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 [37]:

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

Out[37]:

	birds	age	visits	priority
0	Cranes	3.5	2	yes
1	Cranes	4.0	4	yes
5	Cranes	3.0	4	no
9	spoonbills	4.0	2	no

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

In [38]:

```
gr=df.groupby('birds')
cr=gr.get_group('Cranes')
cr['visits'].sum()
```

Out[38]:

```
12
```

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

In [39]:

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

Out[39]:

```
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 [40]:

```
datak = {'birds': 'Parrot', 'age': 4, 'visits': 10, 'priority': 'no'}
app=df.append(datak, ignore_index=True)
print(app)
print("-"*36)
drp=df.drop([9])
print(drp)
```

```
         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      Parrot  4.0      10       no
------------------------------------
        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
```

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

In [41]:

```
gr.count()
```

Out[41]:

	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 [42]:

```
age=df.sort_values('age', ascending =False)
print(age)
print("*"*35)
visits=df.sort_values('visits')
print(visits)
```

```
        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
***********************************
        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 [43]:

```
df.priority.map(dict(yes=1, no=0))
```

Out[43]:

```
0    1
1    1
2    0
3    1
4    0
5    0
6    0
7    1
8    0
9    0
Name: priority, dtype: int64
```

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

In [44]:

```
df.birds.map(lambda x: 'trumpeters' if x =='Cranes' else x)
```

Out[44]:

```
0    trumpeters
1    trumpeters
2       plovers
3    spoonbills
4    spoonbills
5    trumpeters
6       plovers
7    trumpeters
8    spoonbills
9    spoonbills
Name: birds, dtype: object
```

In [45]:

```
#Reference:
#https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_value
s.html
#https://brohrer.github.io/dataframe_indexing.html
#https://www.geeksforgeeks.org/python-pandas-dataframe-append/
#https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html
```