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 [5]:

```
#Data
import numpy as np
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 [124]:

```
import pandas as pd
pandas.DataFrame(data=None, index=labels, columns=None, dtype=None, copy=False)
Output= pd.DataFrame(data['birds'],columns=['birds'],index=labels)
Output
```

Out[124]:

	birds
a	Cranes
b	Cranes
c	plovers
d	spoonbills
e	spoonbills
f	Cranes
g	plovers
h	Cranes
i	spoonbills
j	spoonbills

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

In [125]:

```
df = pd.DataFrame(data,index=labels)
print("Summary of the basic information about this DataFrame and its data:")
print(df.info())
```

```
Summary of the basic information about this DataFrame and its 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 [34]:

```
df = pd.DataFrame(data['birds'] ,index=labels,columns=['birds'])
print("First two rows of the  birds data frame:")
print(df.iloc[:2])
```

```
First two rows of the  birds data frame:
    birds
a  Cranes
b  Cranes
```

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

In [128]:

```
df = pd.DataFrame(data)
# selecting two columns  birds and age
df[['birds', 'age']]
```

Out[128]:

	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 [43]:

```
df.loc[[2, 3, 7],['birds', 'age','visits']]
```

Out[43]:

	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 [45]:

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

Out[45]:

	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 [81]:

```
df = pd.DataFrame(data)
print("Rows where age is missing:")
ef=df[df['age'].isnull()]
ef[['birds', 'visits']]
```

```
Rows where age is missing:
```

Out[81]:

	birds	visits
3	spoonbills	4
7	Cranes	2

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

In [86]:

```
print("Rows where the birds is a Cranes and the age is less than 4")
print(df[(df['birds'] == 'Cranes') & (df['age'] < 4)])
```

```
Rows where the birds is a Cranes and the age is less than 4
    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 [85]:

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

```
        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 [122]:

```
df[(df['birds'] == 'Cranes') & (df['visits'] > 0)].sum()
```

Out[122]:

```
birds       CranesCranesCranesCranes
```

```
age                             10.5
visits                            12
priority                           3
dtype: object
```

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

In [101]:

```
df.groupby('birds')['age'].mean()
```

Out[101]:

```
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 [103]:

```
print("Orginal rows:")
print(df)
print("\nAppend a new row:")
df.loc['k'] = [100, 'pranav', 'no', 1546.5]
print("Print all records after insert a new record:")
print(df)
print("\nDelete the new row and display the original  rows:")
df = df.drop('k')
print(df)
```

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

Append a new row:
Print all records after insert a new record:
        birds     age visits priority
0      Cranes     3.5      2      yes
1      Cranes       4      4      yes
2     plovers     1.5      3       no
3  spoonbills     NaN      4      yes
4  spoonbills       6      3       no
5      Cranes       3      4       no
6     plovers     5.5      2       no
7      Cranes     NaN      2      yes
8  spoonbills       8      3       no
9  spoonbills       4      2       no
k         100  pranav     no   1546.5

Delete the new row and display the original  rows:
        birds  age visits priority
0      Cranes  3.5      2      yes
1      Cranes    4      4      yes
2     plovers  1.5      3       no
3  spoonbills  NaN      4      yes
4  spoonbills    6      3       no
5      Cranes    3      4       no
6     plovers  5.5      2       no
7      Cranes  NaN      2      yes
8  spoonbills    8      3       no
```

```
9  spoonbills    4      2       no
```

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

In [129]:

```
df.birds.groupby(df["birds"]).count()
```

Out[129]:

```
birds
Cranes        4
plovers       2
spoonbills    4
Name: birds, 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 [113]:

```
print("Age in decending order")
sortbyage = df.sort_values('age',ascending=False)
print(sortbyage)
print("-----------------------------------")
print("Visits ascending in  order")
sortbyvists = df.sort_values('visits')
print(sortbyvists)
```

```
Age in decending order
        birds  age visits priority
8  spoonbills    8      3       no
4  spoonbills    6      3       no
6     plovers  5.5      2       no
1      Cranes    4      4      yes
9  spoonbills    4      2       no
0      Cranes  3.5      2      yes
5      Cranes    3      4       no
2     plovers  1.5      3       no
3  spoonbills  NaN      4      yes
7      Cranes  NaN      2      yes
-----------------------------------
Visits ascending in  order
        birds  age visits priority
0      Cranes  3.5      2      yes
6     plovers  5.5      2       no
7      Cranes  NaN      2      yes
9  spoonbills    4      2       no
2     plovers  1.5      3       no
4  spoonbills    6      3       no
8  spoonbills    8      3       no
1      Cranes    4      4      yes
3  spoonbills  NaN      4      yes
5      Cranes    3      4       no
```

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

In [114]:

```
print("Orginal rows:")
print(df)
print("\nReplace the priority column values with'yes' should be 1 and 'no' should be 0")
df['priority'] = df['priority'].map({'yes': 1, 'no': 0})
print(df)
```

```
Orginal rows:
        birds  age visits priority
0      Cranes  3.5      2      yes
1      Cranes    4      4      yes
2     plovers  1.5      3       no
```

```
3  spoonbills  NaN       4       yes
4  spoonbills    6       3        no
5      Cranes    3       4        no
6     plovers  5.5       2        no
7      Cranes  NaN       2       yes
8  spoonbills    8       3        no
9  spoonbills    4       2        no

Replace the priority column values with'yes' should be 1 and 'no' should be 0
        birds  age visits  priority
0      Cranes  3.5      2         1
1      Cranes    4      4         1
2     plovers  1.5      3         0
3  spoonbills  NaN      4         1
4  spoonbills    6      3         0
5      Cranes    3      4         0
6     plovers  5.5      2         0
7      Cranes  NaN      2         1
8  spoonbills    8      3         0
9  spoonbills    4      2         0
```

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

In [120]:

```
Output= pd.DataFrame(df.birds.map(lambda x: 'trumpeters' if x=='Cranes' else x) ,columns=['birds'])
Output
```

Out[120]:

	birds
0	trumpeters
1	trumpeters
2	plovers
3	spoonbills
4	spoonbills
5	trumpeters
6	plovers
7	trumpeters
8	spoonbills
9	spoonbills