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              #importing pandas,numpy libraries
import numpy as np

birds_data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plove
rs', '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']}  #creating a data
frame

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']   #inserting the labels into dataframe

df = pd.DataFrame(birds_data , index = labels )
df          #it  displays the dataframe
```

Out[1]:

	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 [34]:

```
df.describe()     #it will gives summary of data frame
```

Out[34]:

	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 [4]:

```
df.head(2)     #head(2) used to print first two rows in data frame
```

Out[4]:

	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 [5]:

```
df [['birds' , 'age']]  #prints rows with birds, age
```

Out[5]:

	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 [3]:

```
df.iloc[[1,2,7] , [0,1,2]]  #iloc is used to retrive the data from particular rows,columns
```

Out[3]:

	birds	age	visits
b	Cranes	4.0	4
c	plovers	1.5	3
h	Cranes	NaN	2

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

In [7]:

```
df.loc[df['visits'] < 4]     #loc meathod access a group of rows and columns by labels or a boolean
array
```

Out[7]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes

	birds	age	visits	priority
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 [8]:

```
df[df['age'].isnull()]     #its prints the rows where age is null
```

Out[8]:

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

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

In [96]:

```
df[(df['birds'] == 'Cranes') & (df['age'] < 4)] #it prints the rows where carnes having the age <4
```

Out[96]:

	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 [10]:

```
df[(df['age'] >= 2) & (df['age'] <= 4)]    #prints rows where age b/w 2 and 4
```

Out[10]:

	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 [77]:

```
a = df.groupby('birds')    # groupby is used group all the birds into one large group
Cranes1 = a.get_group('Cranes')  #get_group is make a group which contain only Cranes
Cranes1['visits'].sum()    #prints total num of visits
```

Out[77]:

```
12
```

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

In [85]:

```
a = df.groupby('birds')
Cranes1 = a.get_group('Cranes')
Cranes1['age'].mean()  #prints mean age of Cranes birds
```

Out[85]:

```
3.5
```

In [86]:

```
plovers1 = a.get_group('plovers')
plovers1['age'].mean()        #prints mean age of plovers birds
```

Out[86]:

```
3.5
```

In [87]:

```
spoonbills1 = a.get_group('spoonbills')
spoonbills1['age'].mean()       #prints mean age of spoonbills birds
```

Out[87]:

```
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 [130]:

```
df.loc['k'] = ['Parrot', 4.0, 4 , 'yes']    #adding new row into dataframe
df
```

Out[130]:

	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	Parrot	4.0	4	yes

In [133]:

```
df1 = df.drop('k')   #Dropping (or) deleting the row
df1
```

Out[133]:

	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

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

In [94]:

```
count = df['birds'].value_counts()
print(count)       #prints diff typ of birds with count
```

```
Cranes         4
spoonbills     4
plovers        2
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 [108]:

```
dec_age = df.sort_values('age',ascending=False)
print(dec_age.head())  #print age in decending order
```

```
        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
```

In [101]:

```
asc_vists = df.sort_values('visits',ascending=True)
print(asc_vists.head())     #visits in ascending order
```

```
        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
```

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

In [28]:

```
df.priority.map(dict(yes=1, no=0))   #replace the values( 'yes' with '1' , 'no' with '0')
```

Out[28]:

```
a    1
b    1
c    0
d    1
e    0
f    0
g    0
```

```
h    1
i    0
j    0
Name: priority, dtype: int64
```

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

In [27]:

```
df['birds'] = df['birds'].replace('Cranes', 'trumpeters')  #in the birds column Cranes renamed
with trumpeters
df['birds']
```

Out[27]:

```
a    trumpeters
b    trumpeters
c       plovers
d    spoonbills
e    spoonbills
f    trumpeters
g       plovers
h    trumpeters
i    spoonbills
j    spoonbills
Name: birds, dtype: object
```

In [ ]:

```

```

In [ ]:

```

```