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 [136]:

```
import pandas as pd
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']}
birds = pd.DataFrame(data,columns=["birds","age","visits","priority"])
birds.head()
```

Out[136]:

	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

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

In [137]:

```
birds.describe()
```

Out[137]:

	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 [138]:

```
birds.iloc[:2]
```

Out[138]:

	birds	age	visits	priority

	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 [139]:

```
df = birds[["birds","age"]]
df
```

Out[139]:

	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 [140]:

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

Out[140]:

	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 [141]:

```
birds.iloc[:][birds["visits"]<4]
```

Out[141]:

	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 [142]:

```
birds.loc[:,["birds","visits"]][birds["age"].isnull() ]
```

Out[142]:

	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 [143]:

```
birds.loc[:][(birds["birds"]=="Cranes") & (birds["age"]<4)]
```

Out[143]:

	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 [144]:

```
birds.loc[:][(birds["age"]>2) & (birds["age"]<=4)]
```

Out[144]:

	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 [145]:

```
cranes_df = birds.loc[:][birds["birds"]=="Cranes"]
total_visits_cranes = cranes_df["visits"].sum(axis=0)
total_visits_cranes
```

Out[145]:

```
12
```

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

In [146]:

```
uniq = birds["birds"].unique()
uniq
cranes_df = birds.loc[:][birds["birds"]=="Cranes"]
plovers_df = birds.loc[:][birds["birds"]=="plovers"]
spoonbills_df = birds.loc[:][birds["birds"]=="spoonbills"]

crane_mean = cranes_df["age"].mean()
plovers_mean = plovers_df["age"].mean()
```

```
spoonbills_mean = spoonbills_df["age"].mean()

crane_mean,plovers_mean,spoonbills_mean
```

Out[146]:

```
(3.5, 3.5, 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 [147]:

```
birds

add_data = {"birds":"Crow","age":2,"visits":3,"priority":"yes"}

birds.append(add_data,ignore_index=True)

birds = birds[birds.birds!="Crow"]
birds
```

Out[147]:

	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

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

In [148]:

```
birds["birds"].value_counts()
```

Out[148]:

```
spoonbills    4
Cranes        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 [149]:

```
birds.sort_values("age",ascending = False)
birds.sort_values("visits",ascending = True)
```

Out[149]:

	birds	age	visits	priority
0	Cranes	3.5	2	yes
6	plovers	5.5	2	no

	birds	age	visits	priority
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 [150]:

```
birds["priority"] = birds["priority"].apply({"yes":1,"no":0}.get)
birds
```

Out[150]:

	birds	age	visits	priority
0	Cranes	3.5	2	1
1	Cranes	4.0	4	1
2	plovers	1.5	3	0
3	spoonbills	NaN	4	1
4	spoonbills	6.0	3	0
5	Cranes	3.0	4	0
6	plovers	5.5	2	0
7	Cranes	NaN	2	1
8	spoonbills	8.0	3	0
9	spoonbills	4.0	2	0

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

In [153]:

```
birds = birds.replace(to_replace = "Cranes",value = "trumpeters")
birds
```

Out[153]:

	birds	age	visits	priority
0	trumpeters	3.5	2	1
1	trumpeters	4.0	4	1
2	plovers	1.5	3	0
3	spoonbills	NaN	4	1
4	spoonbills	6.0	3	0
5	trumpeters	3.0	4	0
6	plovers	5.5	2	0
7	trumpeters	NaN	2	1
8	spoonbills	8.0	3	0
9	spoonbills	4.0	2	0

In [ ]: