

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 [241]:

```
import pandas as pd
import numpy as np
birds=pd.DataFrame({
    "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"]
})
```

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

In [2]:

```
birds
```

Out[2]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
4	spoonbills	6.0	3	no	e
5	Cranes	3.0	4	no	f
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

3. Print the first 2 rows of the birds dataframe

In [5]:

```
birds.head(2)
```

Out[5]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b

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

In [6]:

```
birds[["birds", "age"]]
```

Out[6]:

	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 [27]:

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

Out[27]:

	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 [4]:

```
b=birds[birds['visits']<4]
b
```

Out[4]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
2	plovers	1.5	3	no	c
4	spoonbills	6.0	3	no	e
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

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

In [78]:

```
birds_age = birds[birds['age'].isnull()]
birds_desc = birds_age.loc[:,['birds','visits']]
birds_desc
```

Out[78]:

	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 [89]:

```
birds_age = birds[birds['age']<4]
birds_category = birds_age[birds['birds']=='Cranes']
birds_category
```

C:\Users\user\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

Out[89]:

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

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

In [96]:

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

Out[96]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
4	spoonbills	6.0	3	no	e
5	Cranes	3.0	4	no	f
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

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

In [114]:

```
birds_category = birds[birds['birds']=='Cranes']
print(birds_category)
total_visits = birds_category.visits.sum()
print("=====")
print("total visit:",total_visits)
#total_visits = birds_category[birds_category['visits']]
```

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
5	Cranes	3.0	4	no	f
7	Cranes	NaN	2	yes	h

=====
total visit: 12

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

In [97]:

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

Out[97]:

4.4375

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 [158]:

```
#birds
birds.insert(5,'k',['a','b','c','d','e','f','g','h','i','j'],True)
print("Dataframe after adding new column\n",birds)
del birds['k']
print("DataFrame after column deletion\n",birds)
```

Dataframe after adding new column

	birds	age	visits	priority	labels	k
0	Cranes	3.5	2	yes	a	a
1	Cranes	4.0	4	yes	b	b
2	plovers	1.5	3	no	c	c
3	spoonbills	NaN	4	yes	d	d
4	spoonbills	6.0	3	no	e	e
5	Cranes	3.0	4	no	f	f
6	plovers	5.5	2	no	g	g
7	Cranes	NaN	2	yes	h	h
8	spoonbills	8.0	3	no	i	i
9	spoonbills	4.0	2	no	j	j

DataFrame after column deletion

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
4	spoonbills	6.0	3	no	e
5	Cranes	3.0	4	no	f
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

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

In [178]:

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

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
4	spoonbills	6.0	3	no	e
5	Cranes	3.0	4	no	f
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

Out[178]:

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

14. Sort dataframe (birds) first by the values in the 'age' in descending order, then by the value in the 'visits' column in ascending order.

In [193]:

```
birds_age_desc=birds.sort_values(['age'],axis=0,ascending=False)
print("=====Age in descending order=====")
print(birds_age_desc)
birds_visit_asc=birds.sort_values(['visits'],axis=0,ascending=True)
print("=====Visits in ascending order=====")
print(birds_visit_asc)
```

=====Age in descending order=====

	birds	age	visits	priority	labels
8	spoonbills	8.0	3	no	i
4	spoonbills	6.0	3	no	e
6	plovers	5.5	2	no	g
1	Cranes	4.0	4	yes	b
9	spoonbills	4.0	2	no	j
0	Cranes	3.5	2	yes	a
5	Cranes	3.0	4	no	f
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
7	Cranes	NaN	2	yes	h

=====Visits in ascending order=====

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
9	spoonbills	4.0	2	no	j
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
2	plovers	1.5	3	no	c
4	spoonbills	6.0	3	no	e
8	spoonbills	8.0	3	no	i
5	Cranes	3.0	4	no	f
1	Cranes	4.0	4	yes	b
3	spoonbills	NaN	4	yes	d

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

In [246]:

```
birds1=birds.replace(to_replace=["yes"],value =1)
birds2=birds1.replace(to_replace=["no"],value =0)
birds2
```

Out[246]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	1	a
1	Cranes	4.0	4	1	b
2	plovers	1.5	3	0	c
3	spoonbills	NaN	4	1	d
4	spoonbills	6.0	3	0	e
5	Cranes	3.0	4	0	f
6	plovers	5.5	2	0	g
7	Cranes	NaN	2	1	h
8	spoonbills	8.0	3	0	i
9	spoonbills	4.0	2	0	j

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

In [247]:

```
birds=birds.replace('Cranes','trumpeters')
birds
```

Out[247]:

	birds	age	visits	priority	labels
0	trumpeters	3.5	2	yes	a
1	trumpeters	4.0	4	yes	b
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
4	spoonbills	6.0	3	no	e
5	trumpeters	3.0	4	no	f
6	plovers	5.5	2	no	g
7	trumpeters	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

In []: