

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 [62]:

```
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']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df=pd.DataFrame(data,columns=['birds','age','visits','priority'],index=labels)
```

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

In [63]:

df

Out[63]:

	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

3. Print the first 2 rows of the birds dataframe

In [64]:

df[:2]

Out[64]:

	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 [65]:

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

Out[65]:

	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 [66]:

```
df1=df.loc[['b', 'c', 'g']]
df1[['birds', 'age', 'visits']]
```

Out[66]:

	birds	age	visits
b	Cranes	4.0	4
c	plovers	1.5	3
g	plovers	5.5	2

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

In [67]:

```
df[df.visits!=df.visits.max()]
```

Out[67]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes
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 [68]:

```
"""
```

Referred Google for seeing how to extract only those rows which has NaN value i.e., use of isnull() function

```
df[['birds','visits']][df.age.isnull()==True]
```

Out[68]:

	birds	visits
d	spoonbills	4
h	Cranes	2

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

In [69]:

```
df[df.birds=='Cranes'][df.age<4]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    """Entry point for launching an IPython kernel.
```

Out[69]:

	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 [70]:

```
df[df.age>2][df.age<4]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    """Entry point for launching an IPython kernel.
```

Out[70]:

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

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

In [71]:

```
"""  
Referred google to see the use of sum function  
"""  
total=df['visits'].sum()  
print(total)
```

29

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

In [72]:

```
g=df.groupby('birds')  
print(g.age.mean())
```

```
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 [80]:

```
"""
Referred google for seeing the use of drop command
"""
df1=pd.DataFrame({'birds':'Parrot','age':3.4,'visits':6,'priority':'yes'},columns=['bird
s','age','visits','priority'],index=['k'])
df=pd.concat([df,df1])
df
df=df.drop(['k'],axis=0)
df
```

Out[80]:

	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

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

In [81]:

```
g=df.groupby('birds')
df1=g.count()
df1[['visits']]
```

Out[81]:

	visits
birds	
Cranes	4
plovers	2
spoonbills	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 [82]:

```
"""
Referred google for sort_value function
"""
df.sort_values('age',ascending=False)
df.sort_values('visits')
```

Out[82]:

	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
e	spoonbills	6.0	3	no
i	spoonbills	8.0	3	no
b	Cranes	4.0	4	yes
d	spoonbills	NaN	4	yes
f	Cranes	3.0	4	no

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

In [83]:

```
"""
Referred google for seeing the use of replace function
"""
df['priority']=df['priority'].replace(['yes'],1)
df['priority']=df['priority'].replace(['no'],0)
df
```

Out[83]:

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

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

In [84]:

```
df['birds']=df['birds'].replace(['Cranes'],'trumpeters')
df
```

Out[84]:

	birds	age	visits	priority
a	trumpeters	3.5	2	1
b	trumpeters	4.0	4	1
c	plovers	1.5	3	0
d	spoonbills	NaN	4	1

e	birds	age	visits	priority
f	trumpeters	3.0	4	0
g	plovers	5.5	2	0
h	trumpeters	NaN	2	1
i	spoonbills	8.0	3	0
j	spoonbills	4.0	2	0

In []: