

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 [38]: import pandas as pd  
import numpy as np  
birds_data=pd.DataFrame({'birds': ['Cranes', 'Cranes', 'plovers', 'spoo  
nbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes', 'spoonbills', 'sp  
oonbills'],  
                        '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']},index=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'  
])  
birds_data
```

Out[38]:

	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

	birds	age	visits	priority
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 [39]: `birds_data.describe()`

Out[39]:

	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 [40]: `birds_data.head(2)`

Out[40]:

	birds	age	visits	priority
--	-------	-----	--------	----------

	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 [41]: birds_data[['birds', 'age']]
```

Out[41]:

	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 [42]: #birds_data.iloc[[2,3,7],]
birds_data[['birds', 'age', 'visits']].iloc[[2,3,7]]
```

Out[42]:

	birds	age	visits
c	plovers	1.5	3
d	spoonbills	NaN	4
h	Cranes	NaN	2

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

```
In [43]: #birds_data[birds_data.visits==(birds_data.visits<4)]
birds_data[birds_data.visits<4]
```

Out[43]:

	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 [44]: birds_data[birds_data.age.isnull()]
```

Out[44]:

	birds	age	visits	priority
d	spoonbills	NaN	4	yes

	birds	age	visits	priority
h	Cranes	NaN	2	yes

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

In [45]: `birds_data[(birds_data.birds=='Cranes') & (birds_data.age<4)]`

Out[45]:

	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 [46]: `birds_data[(birds_data.age>=2) & (birds_data.age<=4)]`

Out[46]:

	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 [47]: `#birds_data[birds_data.visits].sum()
#g=birds_data.groupby('birds')
#g.get_group('Cranes').sum()
birds_data.loc[birds_data.birds=='Cranes','visits'].sum()`

Out[47]: 12

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

```
In [49]: g=birds_data.groupby('birds').mean()  
g.age
```

```
Out[49]: 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 [50]: df=pd.DataFrame([{'birds':'Sparrow','age':4.0,'visits':5,'priority':'yes'}],index=['k'])  
#new_data=birds_data.append([{'birds':'Sparrow','age':4.0,'visits':5,'priority':'yes'}],ignore_index=True)  
birds_data=birds_data.append(df,sort=False,ignore_index=False)  
birds_data
```

Out[50]:

	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

	birds	age	visits	priority
h	Cranes	NaN	2	yes
i	spoonbills	8.0	3	no
j	spoonbills	4.0	2	no
k	Sparrow	4.0	5	yes

```
In [51]: #birds_data.drop(birds_data.index['k'])
birds_data.drop('k')
```

Out[51]:

	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 [52]: birds_data.birds.count()
g=birds_data.groupby('birds')
g.birds.count()
```

```
Out[52]: birds
Cranes      4
Sparrow     1
plovers     2
spoonbills  4
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 [53]: birds_data.sort_values('birds',ascending=False)
```

```
Out[53]:
```

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

```
In [54]: birds_data.sort_values('visits')
#OR
#birds_data.sort_values(by='visits')
```


Out[54]:

	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
k	Sparrow	4.0	5	yes

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

```
In [55]: birds_data['priority']=birds_data['priority'].replace(['yes'],[1])
birds_data['priority']=birds_data['priority'].replace(['no'],[0])
birds_data
```

Out[55]:

	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

	birds	age	visits	priority
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
k	Sparrow	4.0	5	1

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

```
In [56]: birds_data['birds']=birds_data['birds'].replace(['Cranes'], ['trumpeters'])
birds_data
```

Out[56]:

	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	spoonbills	6.0	3	0
f	trumpeters	3.0	4	0
g	plovers	5.5	2	0
h	trumpeters	NaN	2	1
i	spoonbills	8.0	3	0

	birds	age	visits	priority
j	spoonbills	4.0	2	0
k	Sparrow	4.0	5	1