

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']
```

```
In [1]: import numpy as np
import pandas as pd
```

```
In [2]: 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']}
```

```
In [3]: 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 [4]: data = pd.DataFrame(data,index=labels)
```

```
In [5]: data
```

```
Out[5]:
```

	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 [6]: summary = data.describe()
```

```
In [7]: summary
```

```
Out[7]:
```

	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 [8]: rows_2 = data.head(2)
```

```
In [9]: rows_2
```

```
Out[9]:
```

	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 [10]: data_b_a = data[['birds','age']]
```

```
In [11]: data_b_a
```

```
Out[11]:
```

	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 [12]: filter_3 = data[['birds','age','visits']].iloc[[2,3,7]]
```

```
In [13]: filter_3
```

```
Out[13]:
```

	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 [14]: filter_4 = data[data['visits']<4]
```

```
In [15]: filter_4
```

```
Out[15]:
```

	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 [16]: filter_5 = data[data['age'].isnull()].filter(data[['birds','visits']])
```

```
In [17]: filter_5
```

```
Out[17]:
```

	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 [18]: filter_6 = data[(data['birds']=='Cranes') & (data['age']<4)]
```

```
In [19]: filter_6
```

```
Out[19]:
```

	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 [20]: filter_7 = data[(data['age']>2) & (data['age']<=4)]
```

```
In [21]: filter_7
```

```
Out[21]:
```

	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 [22]: filter_8 = data[data['birds']=='Cranes']['visits'].sum()
```

```
In [23]: filter_8
```

```
Out[23]: 12
```

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

```
In [24]: filter_9 = data[['birds','age']].groupby(['birds']).mean()
```

```
In [25]: filter_9
```

```
Out[25]:
```

	age
birds	
Cranes	3.5
plovers	3.5
spoonbills	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 [26]: row = pd.DataFrame({'birds':'Parrot','age':3,'visits':6,'priority':'no'},index=['k'])
data = data.append(row)
data
```

```
Out[26]:
```

	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	3.0	6	no

```
In [27]: data = data.drop(index='k')
```

```
In [28]: data
```

```
Out[28]:
```

	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 [29]: count_birds = data.groupby(data['birds']).size()
```

```
In [30]: count_birds
```

```
Out[30]:
```

birds	
Cranes	4
plovers	2
spoonbills	4
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 [31]: data = data.sort_values(by = ['age','visits'],ascending = [False,True])
```

```
In [32]: data
```

```
Out[32]:
```

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

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

```
In [33]: data2 = data
data2 = data2.replace(to_replace = ['no','yes'],value = [0,1])
data2
```

```
Out[33]:
```

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

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

```
In [34]: data2 = data2.replace(to_replace = 'Cranes',value = 'trumpeters')
data2
```

```
Out[34]:
```

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