

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 [36]: import pandas as pd
import numpy as np

data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cr
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

birds = pd.DataFrame(data, index=labels)
print(birds)
```

	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 [32]: print(birds.info())

<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
birds      10 non-null object
age        8 non-null float64
visits     10 non-null int64
priority   10 non-null object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None
```

**3. Print the first 2 rows of the birds dataframe **

```
In [33]: print(birds.head(2))
```

```
      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 [34]: print(birds[['birds', 'age']])
```

```
      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 [37]: #birds.iloc[[2,3,7], [0, 1, 2]]
birds[['birds', 'age', 'visits']].iloc[[2,3,7]]
```

Out[37]:

	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 [38]: birds[birds.visits < 4]
```

Out[38]:

	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 [39]: birds[['birds', 'visits']][birds.age.isnull()]
```

```
Out[39]:
```

	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 [40]: birds[(birds.birds == 'Cranes') & (birds.age < 4)]
```

```
Out[40]:
```

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

```
Out[41]:
```

	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 [42]: birds[birds.birds == 'Cranes'].visits.sum()
```

```
Out[42]: 12
```

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

```
In [50]: import numpy as np
#birds['age'] = birds['age'].map({np.NaN : 0})
birds.loc[:, 'age'].groupby(birds.birds).mean()
```

```
Out[50]: birds
plovers      3.5
spoonbills   6.0
trumpeters   3.5
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 [44]: #idea from https://stackoverflow.com/questions/16824607/pandas-appending-a-row-to
addRow = pd.Series({'birds':'Cranes', 'age':2.5, 'visits':4, 'priority': 'yes'},
birds = birds.append(addRow)

print(birds)
print('\n \n')
# idea for deleting rows taken from https://www.shanelynn.ie/using-pandas-dataframe

birds = birds.drop('k', axis=0)
print(birds)
```

	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	Cranes	2.5	4	yes

	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 [45]: birds.groupby('birds').size()
```

```
Out[45]: 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 [46]: birds.sort_values(by=['age', 'visits'], na_position='first', ascending=[False, True])
```

```
Out[46]:
```

	birds	age	visits	priority
h	Cranes	NaN	2	yes
d	spoonbills	NaN	4	yes
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

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

```
In [47]: birds['priority'] = birds['priority'].map({'yes' : 1, 'no' : 0})
print(birds)
```

	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 [48]: #birds['birds'] =birds['birds'].replace('Cranes' , 'trumpeters')

# checked this while trying to solve the given question below :
# https://www.edureka.co/community/43220/how-to-change-update-cell-value-in-python

birds.at[birds.birds == 'Cranes', 'birds'] = 'trumpeters'

print(birds)
```

	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
j	spoonbills	4.0	2	0

In []: