

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 [2]: # do common operations here
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, index=labels)

df
```

Out[2]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes
b	Cranes	4.0	4	yes
c	plovers	1.5	3	no

	birds	age	visits	priority
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

1. Create a DataFrame birds from this dictionary data which has the index labels.

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

```
dfBirds
```

Out[3]:

	0
a	Cranes
b	Cranes
c	plovers
d	spoonbills
e	spoonbills
f	Cranes
g	plovers
h	Cranes
i	spoonbills

	0
j	spoonbills

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

In [4]: `dfBirds.describe()`

Out[4]:

	0
count	10
unique	3
top	Cranes
freq	4

3. Print the first 2 rows of the birds dataframe

In [5]: `dfBirds[:2]`

Out[5]:

	0
a	Cranes
b	Cranes

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

In [6]: `df[['birds', 'age']]`

Out[6]:

	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 [7]: dfNew = df[['birds', 'age', 'visits']]
        pd.concat([dfNew[2:4:], dfNew[7:8:]])
```

Out[7]:

	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 [8]: df[df.visits < 4]
```

Out[8]:

	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 [9]: dfNew = df[df.age.isnull()]  
dfNew[['birds', 'visits']]
```

Out[9]:

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

Out[10]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes

	birds	age	visits	priority
f	Cranes	3.0	4	no

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

```
In [11]: df[(df.age >= 2) & (df.age <= 4)]
```

Out[11]:

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

Out[12]: 12

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

```
In [13]: g = df.groupby('birds')
g.age.mean()
```

Out[13]: 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 [14]: print(df)

# create new df with sample values
df1 = pd.DataFrame({
    'birds': ['foo'],
    'age' : 1.0,
    'visits': 2.0,
    'priority': ['no']
}, index=['k'])

# append both data frames
dfNew = pd.concat([df, df1])

print ('\n')
print( '*****' )
print(dfNew)

# dfNew.loc['k']

dfNew = dfNew[:-1]

print ('\n')
print( '*****' )
print(dfNew)
```

	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

```
*****
      birds  age  visits  priority
a      Cranes  3.5    2.0      yes
b      Cranes  4.0    4.0      yes
c     plovers  1.5    3.0       no
d  spoonbills  NaN    4.0      yes
e  spoonbills  6.0    3.0       no
f      Cranes  3.0    4.0       no
g     plovers  5.5    2.0       no
h      Cranes  NaN    2.0      yes
i  spoonbills  8.0    3.0       no
j  spoonbills  4.0    2.0       no
k         foo  1.0    2.0       no
```

```
*****
      birds  age  visits  priority
a      Cranes  3.5    2.0      yes
b      Cranes  4.0    4.0      yes
c     plovers  1.5    3.0       no
d  spoonbills  NaN    4.0      yes
e  spoonbills  6.0    3.0       no
f      Cranes  3.0    4.0       no
g     plovers  5.5    2.0       no
h      Cranes  NaN    2.0      yes
i  spoonbills  8.0    3.0       no
j  spoonbills  4.0    2.0       no
```

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

```
In [15]: countBirds = {}
         for birds, birds_def in g:
             countBirds[birds] = birds_def.birds.count()

         countBirds
```

```
Out[15]: {'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 [16]: df.sort_values(['age', 'visits'], ascending=[0, 1])
```

Out[16]:

	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 [19]: dfNew = df.copy()

dfNew.priority.replace(['yes', 'no'], [1, 0], inplace=True)

dfNew
```

Out[19]:

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

	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 [21]: dfNew2 = df.copy()
dfNew2.birds.replace(['Cranes'], ['trumpeters'], inplace=True)
dfNew2
```

Out[21]:

	birds	age	visits	priority
a	trumpeters	3.5	2	yes
b	trumpeters	4.0	4	yes
c	plovers	1.5	3	no
d	spoonbills	NaN	4	yes
e	spoonbills	6.0	3	no

	birds	age	visits	priority
f	trumpeters	3.0	4	no
g	plovers	5.5	2	no
h	trumpeters	NaN	2	yes
i	spoonbills	8.0	3	no
j	spoonbills	4.0	2	no