

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']
```

```
#initializing pandas and numpy as it is gonna be used everywhere
```

```
import pandas as pd
```

```
import numpy as np
```

```
#initializing data
```

```
data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Crane
```

```
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.

```
birds = pd.DataFrame(data['birds'], labels)
```

```
print(birds)
```

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

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

```
# Considering that basic information here means describe method
```

```
df = pd.DataFrame(data)
```

```
print(df.describe())
```

```
      count      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

```
birds_rows = df['birds'][:2]
print(birds_rows)

0    Cranes
1    Cranes
Name: birds, dtype: object
```

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

```
birds_age_rows=df[['birds','age']]
print(birds_age_rows)
```

	birds	age
0	Cranes	3.5
1	Cranes	4.0
2	plovers	1.5
3	spoonbills	NaN
4	spoonbills	6.0
5	Cranes	3.0
6	plovers	5.5
7	Cranes	NaN
8	spoonbills	8.0
9	spoonbills	4.0

5. select [2, 3, 7] rows and in columns ['birds', 'age', 'visits']

```
birds_age_visits_rows= df[['birds','age','visits']]
birds_age_visits_rows = pd.DataFrame(birds_age_visits_rows.values[[2,3,7]],columns=
print(birds_age_visits_rows)
```

	birds	age	visits
0	plovers	1.5	3
1	spoonbills	NaN	4
2	Cranes	NaN	2

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

```
#Considering all columns have to be returned with visits less than 4
df[df['visits']<4]
```

	birds	age	visits	priority
0	Cranes	3.5	2	yes

7. select the rows with columns ['birds', 'visits'] where the age is missing i.e NaN

```
4 spoonbills 6.0 3 no
```

```
df[df['age'].isna()][['birds','visits']]
```

	birds	visits
3	spoonbills	4
7	Cranes	2

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

```
#Considering nan values is also ignored in age column
crane_rows=df.groupby('birds').get_group('Cranes')
print(crane_rows[crane_rows['age']<4])
```

	birds	age	visits	priority
0	Cranes	3.5	2	yes
5	Cranes	3.0	4	no

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

```
df[(df['age']>=2) & (df['age']<=4)]
```

	birds	age	visits	priority
0	Cranes	3.5	2	yes
1	Cranes	4.0	4	yes
5	Cranes	3.0	4	no
9	spoonbills	4.0	2	no

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

```
crane_rows=df.groupby('birds').get_group('Cranes')
print(crane_rows['visits'].sum())
```

12

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

```
birds_rows = df.groupby('birds').mean()
print(birds_rows['age'])
```

```
print(birds_rows[ age ])

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.

```
#Reference taken from- https://www.geeksforgeeks.org/python-delete-rows-columns-fro
#Adding the row
df.loc[len(df.index)] = [ 'Crow',5.0,3,"yes" ]
print("Adding")
print(df)

#Removing row
print("Removing")
df.drop(index=len(df.index)-1,inplace=True)
print(df)
```

```
Adding
   birds  age  visits  priority
0  Cranes  3.5      2      yes
1  Cranes  4.0      4      yes
2  plovers  1.5      3       no
3  spoonbills  NaN      4      yes
4  spoonbills  6.0      3       no
5  Cranes  3.0      4       no
6  plovers  5.5      2       no
7  Cranes  NaN      2      yes
8  spoonbills  8.0      3       no
9  spoonbills  4.0      2       no
10   Crow  5.0      3      yes
Removing
   birds  age  visits  priority
0  Cranes  3.5      2      yes
1  Cranes  4.0      4      yes
2  plovers  1.5      3       no
3  spoonbills  NaN      4      yes
4  spoonbills  6.0      3       no
5  Cranes  3.0      4       no
6  plovers  5.5      2       no
7  Cranes  NaN      2      yes
8  spoonbills  8.0      3       no
9  spoonbills  4.0      2       no
```

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

```
print(df['birds'].value_counts())

spoonbills    4
Cranes        4
```

```
plovers          2
Name: birds, dtype: int64
```

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.

```
#Descending by age
bird_rows=df.sort_values('age',ascending=False)
print(bird_rows['birds'])
```

```
#Ascending by visits
bird_rows=df.sort_values('visits')
print(bird_rows['birds'])
```

```
8    spoonbills
4    spoonbills
6     plovers
1     Cranes
9    spoonbills
0     Cranes
5     Cranes
2     plovers
3    spoonbills
7     Cranes
Name: birds, dtype: object
0     Cranes
6     plovers
7     Cranes
9    spoonbills
2     plovers
4    spoonbills
8    spoonbills
1     Cranes
3    spoonbills
5     Cranes
Name: birds, dtype: object
```

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

```
#Reference - https://www.geeksforgeeks.org/replace-the-column-contains-the-values-y
df['priority'] = df['priority'].map(
    {'yes':1 , 'no':0})
print(df)
```

	birds	age	visits	priority
0	Cranes	3.5	2	1
1	Cranes	4.0	4	1
2	plovers	1.5	3	0
3	spoonbills	NaN	4	1
4	spoonbills	6.0	3	0
5	Cranes	3.0	4	0
6	plovers	5.5	2	0

7	Cranes	NaN	2	1
8	spoonbills	8.0	3	0
9	spoonbills	4.0	2	0

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

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

	birds	age	visits	priority
0	trumpeters	3.5	2	1
1	trumpeters	4.0	4	1
2	plovers	1.5	3	0
3	spoonbills	NaN	4	1
4	spoonbills	6.0	3	0
5	trumpeters	3.0	4	0
6	plovers	5.5	2	0
7	trumpeters	NaN	2	1
8	spoonbills	8.0	3	0
9	spoonbills	4.0	2	0