

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

data = {'birds'      : ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spo
onbills', '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', 'yes
', 'no', 'no']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
birds = pd.DataFrame(data, index = labels)
birds
```

Out[104]:

| | 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 [105]: `birds.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   birds       10 non-null     object
1   age         8 non-null      float64
2   visits      10 non-null     int64
3   priority    10 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

3. Print the first 2 rows of the birds dataframe

In [106]: `birds.head(2)`

Out[106]:

| | 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 [107]: `birds[['birds', 'age']]`

Out[107]:

| | 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 [108]: birds[['birds', 'age', 'visits']].iloc[[2, 3, 7]]
```

```
Out[108]:
```

| | 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 [109]: birds[birds['visits'] < 4]
```

```
Out[109]:
```

| | 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 [110]: birds[birds.age.isnull()][['birds', 'visits']]
```

```
Out[110]:
```

| | 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 [111]: birds[(birds['birds'] == 'Cranes') & (birds['age'] < 4)]
```

```
Out[111]:
```

| | 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 [112]: birds[(birds['age'] <= 4) & (birds['age'] >= 2)]
```

```
Out[112]:
```

| | 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 [113]: birds[birds['birds'] == 'Cranes'][['visits']].sum()
```

```
Out[113]: visits      12
dtype: int64
```

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

```
In [114]: birds.groupby(['birds']).mean()[['age']]
```

```
Out[114]:
```

| | 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 [115]: new_data = {'birds'      : ['new_bird'],
                    'age'         : [3],
                    'visits'      : [2],
                    'priority'    : ['yes']}

df2 = pd.DataFrame(new_data, index=['k'])
birds = birds.append(df2)
print ("Data frame after inserting new row: \n", birds, "\n\n")
birds = birds.drop('k')
print ("dataframe after dropping the inserted row: \n", birds)
```

```
Data frame after inserting new row:
      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   new_bird  3.0      2      yes
```

```
dataframe after dropping the inserted row:
      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 [121]: birds.groupby(['birds']).size()
```

```
Out[121]: birds
Cranes      4
plovers     2
spoonbills  4
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.

```
In [122]: birds.sort_values(by=[ 'age' ], ascending=False).sort_values(by=[ 'visits' ])
```

Out[122]:

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

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

```
In [123]: birds['priority'] = birds['priority'].map({'yes':1 , 'no':0})
birds
```

Out[123]:

| | 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 [125]: birds["birds"].replace({"Cranes": "Trumpeters"}, inplace=True)  
birds
```

Out[125]:

| | 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 |