

**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 sys  
!{sys.executable} -m pip install pandas
```

```
Requirement already satisfied: pandas in /srv/conda/lib/python3.7/site-  
packages (0.24.2)  
Requirement already satisfied: python-dateutil>=2.5.0 in /srv/conda/li  
b/python3.7/site-packages (from pandas) (2.8.0)  
Requirement already satisfied: numpy>=1.12.0 in /srv/conda/lib/python3.  
7/site-packages (from pandas) (1.16.2)  
Requirement already satisfied: pytz>=2011k in /srv/conda/lib/python3.7/  
site-packages (from pandas) (2018.9)  
Requirement already satisfied: six>=1.5 in /srv/conda/lib/python3.7/sit  
e-packages (from python-dateutil>=2.5.0->pandas) (1.12.0)
```

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

```
In [2]: import pandas as pd  
import numpy as np  
  
# dictionary:  
data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbi  
lls', '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       |
| 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 [3]: print("Shape:", df.shape)
print("*****")
print("Columns:", df.columns)
print("*****")
print("df values")
print(df.count())
print("*****")
print(df['birds'].value_counts())
```

```
Shape: (10, 4)
*****
Columns: Index(['birds', 'age', 'visits', 'priority'], dtype='object')
```

```
*****
df values
birds      10
age        8
visits     10
priority   10
dtype: int64
*****
Cranes      4
spoonbills  4
plovers     2
Name: birds, dtype: int64
```

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

In [4]: `df.head(2)`

Out[4]:

|          | birds  | age | visits | priority |
|----------|--------|-----|--------|----------|
| <b>a</b> | Cranes | 3.5 | 2      | yes      |
| <b>b</b> | Cranes | 4.0 | 4      | yes      |

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

In [5]: `df[['birds', 'age']]`

Out[5]:

|          | birds      | age |
|----------|------------|-----|
| <b>a</b> | Cranes     | 3.5 |
| <b>b</b> | Cranes     | 4.0 |
| <b>c</b> | plovers    | 1.5 |
| <b>d</b> | spoonbills | NaN |
| <b>e</b> | spoonbills | 6.0 |

|          | <b>birds</b> | <b>age</b> |
|----------|--------------|------------|
| <b>f</b> | Cranes       | 3.0        |
| <b>g</b> | plovers      | 5.5        |
| <b>h</b> | Cranes       | NaN        |
| <b>i</b> | spoonbills   | 8.0        |
| <b>j</b> | spoonbills   | 4.0        |

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

```
In [6]: df.iloc[[2,3,7]][['birds', 'age', 'visits']]
```

Out[6]:

|          | <b>birds</b> | <b>age</b> | <b>visits</b> |
|----------|--------------|------------|---------------|
| <b>c</b> | plovers      | 1.5        | 3             |
| <b>d</b> | spoonbills   | NaN        | 4             |
| <b>h</b> | Cranes       | NaN        | 2             |

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

```
In [7]: df[df['visits'] < 4]
```

Out[7]:

|          | <b>birds</b> | <b>age</b> | <b>visits</b> | <b>priority</b> |
|----------|--------------|------------|---------------|-----------------|
| <b>a</b> | Cranes       | 3.5        | 2             | yes             |
| <b>c</b> | plovers      | 1.5        | 3             | no              |
| <b>e</b> | spoonbills   | 6.0        | 3             | no              |
| <b>g</b> | plovers      | 5.5        | 2             | no              |
| <b>h</b> | Cranes       | NaN        | 2             | yes             |
| <b>i</b> | spoonbills   | 8.0        | 3             | no              |

|   | birds      | age | visits | priority |
|---|------------|-----|--------|----------|
| j | spoonbills | 4.0 | 2      | no       |

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

```
In [8]: df[['birds', 'visits']][pd.isna(df['age'])]
```

Out[8]:

|   | 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 [9]: df[df['birds'] == 'Cranes'][df['age'] < 4]
```

```
/srv/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    """Entry point for launching an IPython kernel.
```

Out[9]:

|   | 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 [10]: df[df['age'] >= 2][df['age'] <= 4]
```

```
/srv/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    """Entry point for launching an IPython kernel.
```

Out[10]:

|   | 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 [11]: np.sum(df['visits'][df['birds']=='Cranes'])
```

```
Out[11]: 12
```

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

```
In [12]: g = df.groupby('birds')
print(g['age'].mean())
# g['visits'].mean()
```

```
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 [13]: row = pd.Series({'birds':'Cuckoo','age':2,'visits':3,'priority':'yes'},
name='k')
df = df.append(row)
print(df)
print("*****Removing the row*****")
df = df.drop(['k'], axis = 0)
print(df)
```

|   | 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 | Cuckoo     | 2.0 | 3      | yes      |

\*\*\*\*\*Removing the 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 [14]: df['birds'].value_counts()
```

```
Out[14]: Cranes      4
          spoonbills  4
          plovers     2
          Name: birds, 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 [15]: df = df.sort_values(by=["age", "visits"], ascending=[False, True])
df
```

Out[15]:

|   | 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 [17]: yes_mask = df.index[df['priority'] == 'yes']
no_mask = df.index[df['priority'] == 'no']
df['priority'][yes_mask] = 1
df['priority'][no_mask] = 0
print(df)
```

|   | 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
```

```
/srv/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: Setting
WithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

This is separate from the ipykernel package so we can avoid doing imports until

```
/srv/conda/lib/python3.7/site-packages/ipykernel_launcher.py:4: Setting
WithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

after removing the cwd from sys.path.

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

```
In [19]: df = df.replace("Cranes", "trumpeters")
df
```

Out[19]:

|   | 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        |

|          | birds      | age | visits | priority |
|----------|------------|-----|--------|----------|
| <b>h</b> | trumpeters | NaN | 2      | 1        |
| <b>d</b> | spoonbills | NaN | 4      | 1        |

In [ ]: