**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
```

```
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/22/e6/2d47835f91eb010036be207581fa113fb4e3822ec1b4ba
fb0d3d105fede6/pandas-0.24.2-cp37-cp37m-manylinux1_x86_64.whl (10.1MB)
    100% |████████████████████████████████| 10.1MB 204kB/s eta 0:00:01   15% |█████▏
| 1.5MB 18.2MB/s eta 0:00:01
Collecting pytz>=2011k (from pandas)
  Downloading https://files.pythonhosted.org/packages/61/28/1d3920e4d1d50b19bc5d24398a7cd85cc7b9a75a490570
d5a30c57622d34/pytz-2018.9-py2.py3-none-any.whl (510kB)
    100% |████████████████████████████████| 512kB 3.5MB/s eta 0:00:01
Collecting numpy>=1.12.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/91/e7/6c780e612d245cca62bc3ba8e263038f7c144a96a54f87
7f3714a0e8427e/numpy-1.16.2-cp37-cp37m-manylinux1_x86_64.whl (17.3MB)
    100% |████████████████████████████████| 17.3MB 119kB/s eta 0:00:01                          | 880kB 50.2M
B/s eta 0:00:01    11% |█████                            | 2.1MB 11.0MB/s eta 0:00:0252.0MB/s eta 0:00:01
68% |██████████████████████          | 11.9MB 13.8MB/s eta 0:00:01�         | 12.6MB 56.4MB/s eta 0:00:01M
B/s eta 0:00:01 | 16.8MB 46.2MB/s eta 0:00:01
Requirement already satisfied: python-dateutil>=2.5.0 in /srv/conda/lib/python3.7/site-packages (from pand
as) (2.8.0)
Requirement already satisfied: six>=1.5 in /srv/conda/lib/python3.7/site-packages (from python-dateutil>=
2.5.0->pandas) (1.12.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.16.2 pandas-0.24.2 pytz-2018.9
```

**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', '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 |
| **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 [7]:

```
print(df.describe())

print("*************")
print(df.info())
```

```
             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
*************
<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 [8]:

```
df.head(2)
```

Out[8]:

| | 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 [9]:

```
df[['birds','age']]
```

Out[9]:

| | 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 [10]:

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

Out[10]:

| | 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 [11]:

```
df[df['visits'] < 4]
```

Out[11]:

|   | 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 [12]:

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

Out[12]:

|   | 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 [13]:

```
df[df['birds'] == 'Cranes'][df['age'] < 4]
```

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

Out[13]:

|   | 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 [14]:

```
df[df['age'] >= 2][df['age'] <= 4]
```

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

Out[14]:

|   | 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 [15]:

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

Out[15]:

```
12
```

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

In [16]:

```
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 [17]:

```
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 [18]:

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

Out[18]:

```
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.

In [19]:

```
df = df.sort_values(by=["age","visits"], ascending=[False, True])
df
```

Out[19]:

| | 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 [20]:

```
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: SettingWithCopyWarning:
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-v
iew-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: SettingWithCopyWarning:
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-v
iew-versus-copy
  after removing the cwd from sys.path.
```

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

In [21]:

```
df = df.replace("Cranes","trumpeters")
df
```

Out[21]:

| | 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 |
| **h** | trumpeters | NaN | 2 | 1 |
| **d** | spoonbills | NaN | 4 | 1 |

In [ ]:

```

```