

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 [1]:

```
# Create a dataframe from data dictionary and with index from labels list
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']

birds = pd.DataFrame(data, index = labels)
birds
```

Out[1]:

	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 [2]:

```
# use describe function to display summary  
birds.describe()
```

Out[2]:

	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

In [3]:

```
birds.head(2)
```

Out[3]:

	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 [4]:

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

Out[4]:

	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 [5]:

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

Out[5]:

	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 [6]:

```
birds[birds['visits'] < 4]
```

Out[6]:

	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

Credit to check Nan values : <https://datatofish.com/check-nan-pandas-dataframe/>
[\(https://datatofish.com/check-nan-pandas-dataframe/\)](https://datatofish.com/check-nan-pandas-dataframe/)

In [7]:

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

Out[7]:

	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 [8]:

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

Out[8]:

	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 [9]:

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

Out[9]:

	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 [10]:

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

Out[10]:

```
visits    12
dtype: int64
```

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

In [11]:

```
# use group by and mean()
birds_g = birds.groupby('birds')
birds_g.mean()['age']
```

Out[11]:

	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 [12]:

```

plovers = pd.DataFrame({'birds':'plovers','age':4.25,'visits':6,'priority':'no'}, index
=['k'])
birds = birds.append(plovers)
print('After adding new row k')
print(birds)

# use drop to delete a row
birds.drop('k',inplace=True)
print('After dropping new row k')
print(birds)

```

After adding new row k

	birds	age	visits	priority
a	Cranes	3.50	2	yes
b	Cranes	4.00	4	yes
c	plovers	1.50	3	no
d	spoonbills	NaN	4	yes
e	spoonbills	6.00	3	no
f	Cranes	3.00	4	no
g	plovers	5.50	2	no
h	Cranes	NaN	2	yes
i	spoonbills	8.00	3	no
j	spoonbills	4.00	2	no
k	plovers	4.25	6	no

After dropping new row k

	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)

Credit : <https://stackoverflow.com/questions/47320572/pandas-groupby-and-count>
<https://stackoverflow.com/questions/47320572/pandas-groupby-and-count>

In [13]:

```

# use value counts
birds.value_counts(['birds'])

```

Out[13]:

```

birds
spoonbills    4
Cranes        4
plovers       2
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.

Credit : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.html
(https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.html)

In [14]:

```
# use sort_values for sorting
birds.sort_values(['age', 'visits'], ascending = [False, True])
```

Out[14]:

	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

Credit : <https://www.delftstack.com/howto/python-pandas/pandas-replace-values-in-column/>
(<https://www.delftstack.com/howto/python-pandas/pandas-replace-values-in-column/>)

In [15]:

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

Out[15]:

	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'.

Credit : <https://www.delftstack.com/howto/python-pandas/pandas-replace-values-in-column/>
[\(https://www.delftstack.com/howto/python-pandas/pandas-replace-values-in-column/\)](https://www.delftstack.com/howto/python-pandas/pandas-replace-values-in-column/)

In [16]:

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

Out[16]:

	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