

```
In [194]: ***Consider the following Python dictionary data and Python List labels:**
import numpy as np
import pandas as pd
import statistics

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 [195]: df = pd.DataFrame(data , index=labels)
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

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

```
In [196]: print(df.info())

<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 [197]: print(df.iloc[:2])
```

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

```
Out[198]:
```

	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 [199]: rows = ['c','d','h']
          columns = ['birds','age','visits']
          df.loc[rows,columns]
```

```
Out[199]:
```

	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 [200]: `df.loc[df.visits<4]`

Out[200]:

	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 [235]: `g = df[['birds','visits']]`
`g[df['age'].isnull()]`

Out[235]:

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

Out[201]:

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

```
Out[202]:
```

	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 [219]: g = df[(df['birds'] == 'Cranes')].sum()
print('Total Number of visits of the bird Cranes: ',g['visits'])
g
```

Total Number of visits of the bird Cranes: 12

```
Out[219]: birds      CranesCranesCranesCranes
age                                10.5
visits                             12
priority                           3
dtype: object
```

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

```
In [203]: df.mean(axis=1)
```

```
Out[203]: a    2.75
b    4.00
c    2.25
d    4.00
e    4.50
f    3.50
g    3.75
h    2.00
i    5.50
j    3.00
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 [204]: df.append({'birds':'k', 'age':'2', 'visits':'3', 'priority':'yes'}, ignore_index=True)
```

Out[204]:

	birds	age	visits	priority
0	Cranes	3.5	2	yes
1	Cranes	4	4	yes
2	plovers	1.5	3	no
3	spoonbills	NaN	4	yes
4	spoonbills	6	3	no
5	Cranes	3	4	no
6	plovers	5.5	2	no
7	Cranes	NaN	2	yes
8	spoonbills	8	3	no
9	spoonbills	4	2	no
10	k	2	3	yes

```
In [205]: delete_row = df[df.iloc[:,1]==10].index
df = df.drop(delete_row)
df
```

Out[205]:

	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 [206]: count = df['birds'].value_counts()
count
```

```
Out[206]: 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 [207]: df.sort_values(by=['age'], ascending = False)
```

```
Out[207]:
```

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

```
In [208]: df.sort_values(by=['age'])
```

```
Out[208]:
```

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

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

```
In [209]: df['priority'] = df['priority'].map({'yes': 1, 'no': 0})
print(df)
```

	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 [211]: df.replace(to_replace=['Cranes'],value='trumpeters',method='ffill')
```

```
Out[211]:
```

	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