

```

import numpy as np
import pandas as pd

***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', 'no',
                     'yes', 'no', 'no']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

# (1) Creating a DataFrame Birds from the Dictionary and indexing labels.

df = pd.DataFrame(data, columns=["birds", "age", "visits",
                                "priority"])
print(df)

```

	birds	age	visits	priority
0	Cranes	3.5	2	yes
1	Cranes	4.0	4	yes
2	plovers	1.5	3	no
3	spoonbills	NaN	4	yes
4	spoonbills	6.0	3	no
5	Cranes	3.0	4	no
6	plovers	5.5	2	no
7	Cranes	NaN	2	yes
8	spoonbills	8.0	3	no
9	spoonbills	4.0	2	no

```

# (2) Displaying the summary of the basic information about birds DataFrame and its Data.

```

```

df.describe()

```

	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) The first two rows of the Birds DataFrame.

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

	birds	age	visits	priority
0	Cranes	3.5	2	yes
1	Cranes	4.0	4	yes

(4) Printing all the rows only 'birds and 'age columns from the DataFrame

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

	birds	age
0	Cranes	3.5
1	Cranes	4.0
2	plovers	1.5
3	spoonbills	NaN
4	spoonbills	6.0
5	Cranes	3.0
6	plovers	5.5
7	Cranes	NaN
8	spoonbills	8.0
9	spoonbills	4.0

(5) Selecting [2,3,7] rows and in columns ['birds','age','visits']

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

	birds	age	visits
2	plovers	1.5	3
3	spoonbills	NaN	4
7	Cranes	NaN	2

*# (6). select the rows where the number of visits is less than 4***

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

	birds	age	visits	priority
0	Cranes	3.5	2	yes
2	plovers	1.5	3	no
4	spoonbills	6.0	3	no
6	plovers	5.5	2	no
7	Cranes	NaN	2	yes
8	spoonbills	8.0	3	no
9	spoonbills	4.0	2	no

*# (7) selecting the rows with columns ['birds', 'visits'] where the age is missing i.e NaN***

```
df = pd.DataFrame(data, columns = ['birds','age','visits'])  
df[df['age'].isnull()]
```

	birds	age	visits
3	spoonbills	NaN	4
7	Cranes	NaN	2

*# (8). Select the rows where the birds is a Cranes and the age is less than 4***

```
options = ['Cranes']
df[(df['age']<4) & df['birds'].isin(options)]
```

	birds	age	visits	priority
0	Cranes	3.5	2	yes
5	Cranes	3.0	4	no

*# (9). Select the rows the age is between 2 and 4(inclusive)***

```
df[(df['age'] >2 ) & (df['age'] < 5)]
```

	birds	age	visits	priority
0	Cranes	3.5	2	yes
1	Cranes	4.0	4	yes
5	Cranes	3.0	4	no
9	spoonbills	4.0	2	no

(10) The total number of visits in the birds 'Cranes'

```
df['visits'].sum(axis=0)
```

29

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

```
df.groupby(['birds'])['age'].mean()
```

birds	age
Cranes	3.5
plovers	3.5
spoonbills	6.0

Name: age, dtype: float64

(12-a) Adding new row to DataFrame

```
new_row = {'birds':'Pigeon','age':'2.5','visits':'3','priority':'yes'}
```

```
new_df=df.append(new_row, ignore_index=True)
new_df
```

	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	Pigeon	2.5	3	yes

(12-b) Deleting the added row to obtain original DataFrame

```
new_df.drop(labels=10, axis=0)
```

	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

*# (13). Find the number of each type of birds in dataframe (Counts)***

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

```
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.'''

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

	birds	age	visits
8	spoonbills	8.0	3
4	spoonbills	6.0	3
6	plovers	5.5	2
9	spoonbills	4.0	2
1	Cranes	4.0	4
0	Cranes	3.5	2
5	Cranes	3.0	4
2	plovers	1.5	3
7	Cranes	NaN	2
3	spoonbills	NaN	4

```
# (15) Replace the priority column values with 'yes' should be 1 and  
'no' should be 0
```

```
df = pd.DataFrame(data, columns = ['birds','age','visits','priority'])  
df.replace(['yes','no'],[1,0])
```

	birds	age	visits	priority
0	Cranes	3.5	2	1
1	Cranes	4.0	4	1
2	plovers	1.5	3	0
3	spoonbills	NaN	4	1
4	spoonbills	6.0	3	0
5	Cranes	3.0	4	0
6	plovers	5.5	2	0
7	Cranes	NaN	2	1
8	spoonbills	8.0	3	0
9	spoonbills	4.0	2	0

```
# (16) In the 'birds' column, change the 'Cranes' entries to  
'trumpeters'.**
```

```
df = pd.DataFrame(data, columns = ['birds','age','visits','priority'])  
df.replace(['Cranes'],'trumpeters')
```

	birds	age	visits	priority
0	trumpeters	3.5	2	yes
1	trumpeters	4.0	4	yes
2	plovers	1.5	3	no
3	spoonbills	NaN	4	yes
4	spoonbills	6.0	3	no
5	trumpeters	3.0	4	no
6	plovers	5.5	2	no
7	trumpeters	NaN	2	yes
8	spoonbills	8.0	3	no
9	spoonbills	4.0	2	no