

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]: 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']

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

print(data,'\n\n')
```

	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]: print(data.describe(),'\n\n')
```

	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]: print(data.head(2),'\n\n')
```

	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]: print(data[['birds','age']],'\n\n')
```

	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]: print(data[['birds','age','visits']].iloc[[2,3,7]],'\n\n')
```

	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]: print(data[data['visits']<4],'\n\n')
```

	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 [7]: print(data[['birds','visits']][data['age'].isnull()],'\n\n')
```

```
#print(data[['birds','visits']][data['age']=='NaN']) prints a Null table,because of py interpreter
#Whereas, using .isnull() prints the table where values are NaN or missing

#https://datatofish.com/check-nan-pandas-dataframe/
```

	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]: print(data[data['birds']=='Cranes'][data['age']<4],'\n\n')
```

	birds	age	visits	priority
a	Cranes	3.5	2	yes
f	Cranes	3.0	4	no

C:\Users\aniket\AppData\Local\Temp\ipykernel_6536\3809107672.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
print(data[data['birds']=='Cranes'][data['age']<4],'\n\n')

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

```
In [9]: print(data[data['age'].between(2,4)],'\n\n' )
```

```
#https://www.w3resource.com/python-exercises/pandas/python-pandas-data-frame-exercise-10.php
```

	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]: print(data['visits'][data['birds']=='Cranes'].sum(),'\n\n')
```

12

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

```
In [11]: print("plovers:",data['age'][data['birds']=='plovers'].mean())
print('Cranes',data['age'][data['birds']=='Cranes'].mean())
print('spoonbills',data['age'][data['birds']=='spoonbills'].mean())
print('\n')
```

```
#or if for some reasons the names are not known,or the list is long
s=[]
for bird in data['birds']:
    s.append(bird)
s=set(s)
for bird in s:
    print(f"{bird}:{data['age'][data['birds']==str(bird)].mean()}")
print('\n')
```

plovers: 3.5
Cranes 3.5
spoonbills 6.0

plovers:3.5
Cranes: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]: data.loc['k']=['plovers',3.5,5,'yes']
#print(data)
data=data.drop('k')
print(data,'\n\n')
```

```
#https://thispointer.com/python-pandas-how-to-add-rows-in-a-dataframe-using-dataframe-append-loc-iloc/
```

	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 [13]: s=[]
for bird in data['birds']:
    s.append(bird)
s=set(s)
for bird in s:
    print(f"{bird}:{data['birds'][data['birds']==str(bird)].count()}")
print('\n\n')
```

plovers:2
Cranes:4
spoonbills:4

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 [14]: print(data.sort_values(by='age',ascending=False),'\n\n')
print(data.sort_values(by='visits',ascending=True),'\n\n')
```

	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

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

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

```
In [15]: data=data.replace(to_replace='yes',value=1)
data=data.replace(to_replace='no',value=0)
print(data,'\n\n')
```

```
#https://www.geeksforgeeks.org/python-pandas-dataframe-replace/
```

	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 [16]: data=data.replace(to_replace='Cranes',value='trumpeters')
print(data,'\n\n')
```

	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