

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 [198]: import pandas as pd

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

Out[198]:

	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 [167]: #To get number of rows and columns  
bird.shape
```

```
Out[167]: (10, 4)
```

```
In [168]: #TO get name of columns  
bird.columns
```

```
Out[168]: Index(['birds', 'age', 'visits', 'priority'], dtype='object')
```

```
In [169]: #Displays datatypes, no of columns and number of values  
bird.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
```

```
In [170]: bird.describe()
```

```
Out[170]:
```

	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

```
In [171]: #To get max age  
bird['age'].max()
```

```
Out[171]: 8.0
```

```
In [172]: #To get min age  
bird['age'].min()
```

```
Out[172]: 1.5
```

```
In [173]: #Displays number of birds of each type  
bird.birds.value_counts()
```

```
Out[173]: Cranes      4  
          spoonbills  4  
          plovers    2  
          Name: birds, dtype: int64
```

3. Print the first 2 rows of the birds dataframe

In [174]: `bird[1:3]`

Out[174]:

	birds	age	visits	priority
b	Cranes	4.0	4	yes
c	plovers	1.5	3	no

4. Print all the rows with only 'birds' and 'age' columns from the dataframe

In [175]: `bird[['birds', 'age']]`

Out[175]:

	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 [176]: bird.iloc[[2,3,7],[0,1,2]]
```

```
Out[176]:
```

	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 [177]: bird[birds.visits<4]
```

```
Out[177]:
```

	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 [178]: k=bird[bird['age'].isnull()]  
k
```

Out[178]:

	birds	age	visits	priority
d	spoonbills	NaN	4	yes
h	Cranes	NaN	2	yes

```
In [179]: k[['birds','visits']]
```

Out[179]:

	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 [180]: bird[(bird['birds']=="Cranes") & (bird['age']<4)]
```

Out[180]:

	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 [181]: data = bird[(bird['age'] >=2) & (bird['age'] <= 4)]  
data
```

Out[181]:

	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 [199]: k=bird[bird['birds']=="Cranes"]  
k
```

Out[199]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes
b	Cranes	4.0	4	yes
f	Cranes	3.0	4	no
h	Cranes	NaN	2	yes

```
In [200]: total=k['visits'].sum()  
print("Total number of visits of bird Cranes are :",total)
```

Total number of visits of bird Cranes are : 12

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

```
In [183]: b=bird.groupby('birds')
          b['age'].mean()
```

```
Out[183]: 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 [184]: #Appending Row
          k={'birds':'peakcock','age':3,'visits':4,'priority':'yes'}
          bird=bird.append(k,ignore_index=True)
          bird
```

```
Out[184]:
```

	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
10	peakcock	3.0	4	yes


```
In [186]: #Delete Row  
bird.drop(bird.index[10], inplace=True)  
bird
```

Out[186]:

	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

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

```
In [187]: bird.birds.value_counts()
```

```
Out[187]: 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.

```
In [188]: #Descending order by values 'age'  
df1 = bird.sort_values(['age'], ascending=[False])  
df1
```

Out[188]:

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

```
In [189]: #Ascending order by values in 'visits'
df2=df1.sort_values(['visits'],ascending=[True])
df2
```

Out[189]:

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

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

```
In [190]: bird['priority']=bird['priority'].replace(['yes'],1)
bird['priority']=bird['priority'].replace(['no'],0)
bird.head()
```

Out[190]:

	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

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

```
In [191]: bird['birds']=bird['birds'].replace(['Cranes'],'trumpeters')
bird.head()
```

Out[191]:

	birds	age	visits	priority
0	trumpeters	3.5	2	1
1	trumpeters	4.0	4	1
2	plovers	1.5	3	0
3	spoonbills	NaN	4	1
4	spoonbills	6.0	3	0

```
In [ ]: !jupyter nbconvert --to html your_notebook_name.ipynb
```