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 [167]:

```
import numpy as np
import pandas as pd
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[167]:

	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 [168]:

```
print ("Summary of Birds DataFrame is: \n",birds.describe())
print ("Top rows of birds data frame: \n",birds.head())
print ("Last rows of data frame is: \n",birds.tail())
print ("Average age of birds is: \n",birds.mean())
```

```
Summary of Birds DataFrame is:
              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
Top rows of birds data frame:
          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
Last rows of data frame is:
```

```
Last rows of data frame is:
         birds  age  visits priority
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
Average age of birds is:
 age       4.4375
visits    2.9000
dtype: float64
```

3. Print the first 2 rows of the birds dataframe

In [169]:

```
print ("First two rows of birds data frame: \n", birds.iloc[0:2])##another way
```

```
First two rows of birds data frame: 
     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 [170]:

```
print (birds[['birds','age']])
```

```
        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 [171]:

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

Out[171]:

	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 [172]:

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

Out[172]:

	birds	age	visits	priority
a	Cranes	3.5	2	yes

	birds	age	visits	priority
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 [173]:

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

Out[173]:

	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 [174]:

```
birds[birds['birds'] == 'Cranes'][birds['age'] < 4]
```

```
C:\Users\Vemuri Gnanesh\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Boolean
Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

Out[174]:

	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 [175]:

```
birds[birds['age'].between(2,4)]
##Reference: https://www.w3resource.com/python-exercises/pandas/python-pandas-data-frame-exercise-
10.php
```

Out[175]:

	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 [176]:

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

Out[176]:

Out[176]:

```
12
```

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

In [177]:

```
birds_g = birds.groupby('birds')
birds_g.mean()
```

Out[177]:

	age	visits
birds		
Cranes	3.5	3.0
plovers	3.5	2.5
spoonbills	6.0	3.0

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 [178]:

```
birds = birds.append(pd.DataFrame([{'birds':'Parrot','age':4,'visits':3,"priority":1}],index = ['f'
]))
print ("Data Frame after appending:\n {0}".format(birds))
birds = birds.drop(labels = ['f'])
print ("Data Frame after deleting:\n {0}".format(birds))
##Reference : appending https://pandas.pydata.org/pandas-
docs/stable/reference/api/pandas.DataFrame.append.html
## deleting : https://www.geeksforgeeks.org/python-delete-rows-columns-from-dataframe-using-pandas
-drop/
```

```
Data Frame after appending:
    age       birds priority  visits
a   3.5      Cranes      yes       2
b   4.0      Cranes      yes       4
c   1.5     plovers       no       3
d   NaN  spoonbills      yes       4
e   6.0  spoonbills       no       3
f   3.0      Cranes       no       4
g   5.5     plovers       no       2
h   NaN      Cranes      yes       2
i   8.0  spoonbills       no       3
j   4.0  spoonbills       no       2
f   4.0      Parrot        1       3
Data Frame after deleting:
    age       birds priority  visits
a   3.5      Cranes      yes       2
b   4.0      Cranes      yes       4
c   1.5     plovers       no       3
d   NaN  spoonbills      yes       4
e   6.0  spoonbills       no       3
g   5.5     plovers       no       2
h   NaN      Cranes      yes       2
i   8.0  spoonbills       no       3
j   4.0  spoonbills       no       2
```

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

In [179]:

```
birds_g = birds.groupby('birds')
for bir,birds_df in birds_g:
    print ("No of {bird} are {counts}".format(bird = bir, counts = len(birds_df['birds'])))
```

```
No of Cranes are 3
No of plovers are 2
No of spoonbills are 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 [180]:

```
print(birds['age'].sort_values())## Sort() will not work for a series object
print ("----====----"*50)
print(birds['visits'].sort_values(ascending = True)) ## By default descending order if ascending=
True then ascending
## Reference: https://www.geeksforgeeks.org/python-pandas-series-sort_values/
```

```
c    1.5
a    3.5
b    4.0
j    4.0
g    5.5
e    6.0
i    8.0
d    NaN
h    NaN
Name: age, dtype: float64
----====--------====--------====--------====--------====--------====--------====--------====--------=
===--------====--------====--------====--------====--------====--------====--------====--------====--
------====--------====--------====--------====--------====--------====--------====--------====-------
-====--------====--------====--------====--------====--------====--------====--------====--------====
--------====--------====--------====--------====--------====--------====--------====--------====-----
---====--------====--------====--------====--------====--------====--------====--------====----
a    2
g    2
h    2
j    2
c    3
e    3
i    3
b    4
d    4
Name: visits, dtype: int64
```

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

In [181]:

```
birds['priority'][birds['priority'] == 'yes'] = 1
birds['priority'][birds['priority'] == 'no'] = 0
birds
```

```
C:\Users\Vemuri Gnanesh\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```

Out[181]:

	age	birds	priority	visits
a	3.5	Cranes	1	2
b	4.0	Cranes	1	4
c	1.5	plovers	0	3
d	NaN	spoonbills	1	4
e	6.0	spoonbills	0	3
g	5.5	plovers	0	2
h	NaN	Cranes	1	2

	age	birds	priority	visits
i	8.0	spoonbills	0	3
j	4.0	spoonbills	0	2

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

In [182]:

```
birds['birds'][birds['birds'] == 'Cranes'] = 'trumpeters'
birds
```

Out[182]:

	age	birds	priority	visits
a	3.5	trumpeters	1	2
b	4.0	trumpeters	1	4
c	1.5	plovers	0	3
d	NaN	spoonbills	1	4
e	6.0	spoonbills	0	3
g	5.5	plovers	0	2
h	NaN	trumpeters	1	2
i	8.0	spoonbills	0	3
j	4.0	spoonbills	0	2