Practice Questions on Pandas

Example :

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']

In [1]:
```
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']

print("data: \n",data)
print("\nlabels: \n",labels)
```

```
data:
 {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes',
'spoonbills', 'spoonbills'], 'age': [3.5, 4, 1.5, nan, 6, 3, 5.5, 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 [2]:
```
df = pd.DataFrame(data,columns=['birds','age','visits','priority'],index= labels)
df
```

Out[2]:

	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 [3]:
```
df['birds'].describe()
```

Out[3]:
```
count             10
unique             3
top       spoonbills
freq               4
Name: birds, dtype: object
```

3. Print the first 2 rows of the birds dataframe

In [4]:
```
df.iloc[0:2,:]
```

Out[4]:

	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 [5]:
```
df.loc[:,['birds','age']]
```

Out[5]:

	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 [6]:
```
df.loc[df.index[[2,3,7]],['birds', 'age', 'visits']]
```

Out[6]:

	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 [7]:
```
df.loc[df['visits'] < 4,:]
```

Out[7]:

	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 [8]:
```
df.loc[df['age'].isnull(),['birds', 'visits']]
```

Out[8]:

	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 [9]:
```
df.loc[(df['birds'] == "Cranes") & (df['age'] < 4),:]
```

Out[9]:

	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 [10]:
```
df.loc[(df['age'] >= 2) & (df['age'] <= 4),:]
```

Out[10]:

	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 [11]:
```
print("The total number of visits of the bird Cranes : ",df.loc[df['birds'] == "Cranes",:]['visits'].sum())
```

```
The total number of visits of the bird Cranes :  12
```

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

In [12]:
```
df.groupby(['birds'])['age'].mean()
```

Out[12]:
```
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 [13]:
```
row = pd.DataFrame({'birds': 'kingfisher', 'age': 7, 'visits': 5,'priority': 'yes'}, index=['k'])
print("Row 'k' is added to dataframe : \n")
added_row_df = df.append(row)
added_row_df
```

```
Row 'k' is added to dataframe :
```

Out[13]:

	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
k	7.0	kingfisher	yes	5

In [14]:
```
print("Row 'k' is deleted from dataframe : \n")
deleted_row_df = added_row_df.drop('k')
deleted_row_df
```

```
Row 'k' is deleted from dataframe :
```

Out[14]:

	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

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

In [15]:
```
df.groupby(['birds'])['birds'].count()
```

Out[15]:
```
birds
Cranes        4
plovers       2
spoonbills    4
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 [16]:
```
df.sort_values(by='age', ascending=False).sort_values(by='visits', ascending=True)
```

Out[16]:

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

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

In [17]:
```
df['priority'] = df['priority'].apply(lambda x: 1 if x == 'yes' else 0)
df
```

Out[17]:

	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 [18]:
```
df['birds'] = df['birds'].apply(lambda x: 'trumpeters' if x == 'Cranes' else x)
df
```

Out[18]:

	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