

pandas_basics_practice

January 28, 2021

```
In [1]: import numpy as np
import pandas as pd
from functools import reduce
```

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 [2]: data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes',
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 [3]: df = pd.DataFrame(data=data, index=labels)
df
```

```
Out[3]:
```

	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 [4]: df.describe()
```

```
Out[4]:
```

	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 [5]: df.iloc[0:2,:]
```

```
Out[5]:
```

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

```
Out[6]:
```

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

```
Out[7]:
```

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

```
Out[8]:
```

	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 [9]: df[~df['age'].notnull()]
```

```
Out[9]:
```

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

8. Select the rows where the birds is a Cranes and the age is less than 4

```
In [10]: print(df.loc[(df.birds=='Cranes') & (df.age < 4)])
```

	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 [11]: print(df[df['age'].between(2,4,inclusive=True)])
```

	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 [12]: df[df['birds'] == 'Cranes']
```

```
Out[12]:
```

	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

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

```
In [13]: df.groupby(by=['birds'])[['age']].mean()
```

```
Out[13]:
```

	age
birds	
Cranes	3.5
plovers	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 [14]: row= {'birds': 'peacock', 'age': 5, 'visits': 6, 'priority': 'yes'}
df.loc['k'] = ['peacock',5,6,'yes']
df.drop('k')
```

```
Out[14]:
```

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

```
Out[15]: birds
Cranes      4
peacock     1
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(['age','visits'],ascending=[False,True])
```

```
Out[16]:
```

	birds	age	visits	priority
i	spoonbills	8.0	3	no
e	spoonbills	6.0	3	no
g	plovers	5.5	2	no
k	peacock	5.0	6	yes
j	spoonbills	4.0	2	no
b	Cranes	4.0	4	yes
a	Cranes	3.5	2	yes
f	Cranes	3.0	4	no
c	plovers	1.5	3	no
h	Cranes	NaN	2	yes
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'].replace({'yes':1,'no':0})
```

```
Out[17]: a    1
         b    1
         c    0
         d    1
         e    0
         f    0
         g    0
         h    1
         i    0
         j    0
         k    1
         Name: priority, dtype: int64
```

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

```
In [18]: df['birds'].replace({'Cranes' : 'trumpeters'})
```

```
Out[18]: a    trumpeters
         b    trumpeters
         c      plovers
         d  spoonbills
         e  spoonbills
         f    trumpeters
         g      plovers
         h    trumpeters
         i  spoonbills
         j  spoonbills
         k      peacock
         Name: birds, dtype: object
```