

Pandas Practice Questions

January 26, 2022

Let's import necessary packages for this assignment.

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

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.

```
[2]: 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']
}
```

```
[3]: labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
[4]: birds_df = pd.DataFrame(data=data, index=labels)
display(birds_df)
```

	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.

```
[5]: birds_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   birds        10 non-null     object
1   age           8 non-null      float64
2   visits       10 non-null     int64
3   priority     10 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

3. Print the first 2 rows of the birds dataframe.

```
[6]: display(birds_df.head(n=2))
```

	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.

```
[7]: display(birds_df[['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'].

```
[8]: display(birds_df.iloc[[2, 3, 7], [0, 1, 2]])
```

	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.

```
[9]: display(birds_df[birds_df['visits'] < 4])
```

	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.

```
[10]: display(birds_df[birds_df['age'].isnull()][['birds', 'visits']])
```

	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.

```
[11]: display(birds_df[(birds_df['birds'] == 'Cranes') & (birds_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).

```
[12]: display(birds_df[(birds_df['age'] >= 2) & (birds_df['age'] <= 4)])
```

	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.

```
[13]: display(birds_df[birds_df['birds'] == 'Cranes']['visits'].sum())
```

```
visits    12
dtype: int64
```

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

```
[14]: display(birds_df.groupby(by='birds')['age'].mean())
```

```
      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.

```
[15]: df = pd.DataFrame(data=[['plovers', 4.5, 2, 'no']], columns=list(birds_df.
    ↪columns), index=['k'])
display(df)
```

	birds	age	visits	priority
k	plovers	4.5	2	no

```
[16]: birds_df = birds_df.append(other=df)
display(birds_df)
```

	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
k	plovers	4.5	2	no

```
[17]: birds_df = birds_df.drop(labels=['k'])
      display(birds_df)
```

	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).

```
[18]: display(birds_df.groupby(by='birds').count())
```

	age	visits	priority
birds			
Cranes	3	4	4
plovers	2	2	2
spoonbills	3	4	4

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.

```
[19]: age_sort_df = birds_df.sort_values(by=['age'], ascending=False)
      display(age_sort_df)
```

	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

```
[20]: visits_sort_df = birds_df.sort_values(by=['visits'], ascending=True)
      display(visits_sort_df)
```

	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.

```
[21]: birds_df['priority'] = birds_df['priority'].map({'yes': 1, 'no': 0})
      display(birds_df)
```

	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’.

```
[22]: birds_df['birds'] = birds_df['birds'].apply(func=lambda x: 'trumpeters' if x == 'Cranes' else x)
      display(birds_df)
```

	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

End of the file.