pandas_basics_practice

July 30, 2018

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 [113]: import pandas as pd
          import numpy as np

          data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes
                  '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']

          df = pd.DataFrame(data)
          df
```

```
Out[113]:         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
```

1. Create a DataFrame birds from this dictionary data which has the index labels.

```
In [119]: pd.DataFrame(df.birds)
```

```
Out[119]:         birds
          0      Cranes
          1      Cranes
```

```
2       plovers
3    spoonbills
4    spoonbills
5        Cranes
6       plovers
7        Cranes
8    spoonbills
9    spoonbills
```

2. Display a summary of the basic information about birds DataFrame and its data.

```
In [17]: df['birds'].describe()
```

```
Out[17]: count             10
         unique             3
         top       spoonbills
         freq               4
         Name: birds, dtype: object
```

3. Print the first 2 rows of the birds dataframe

```
In [18]: #.loc is used for accesing rows based on index
         #.iloc is used for accesing rows based on row numbers(number of rows)
         pd.DataFrame(df['birds'].iloc[:2])
```

```
Out[18]:     birds
         0  Cranes
         1  Cranes
```

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

```
In [19]: #df[['birds','age']]
         df1 = df.iloc[:,0:2]
         df1
```

```
Out[19]:         birds  age
         0      Cranes  3.5
         1      Cranes  4.0
         2     plovers  1.5
         3  spoonbills  NaN
         4  spoonbills  6.0
         5      Cranes  3.0
         6     plovers  5.5
         7      Cranes  NaN
         8  spoonbills  8.0
         9  spoonbills  4.0
```

5. select [2, 3, 7] rows and in columns ['birds', 'age', 'visits']

```
In [20]: df.iloc[[2, 3, 7]]
```

```
Out[20]:         birds  age  visits priority
         2     plovers  1.5       3       no
         3  spoonbills  NaN       4      yes
         7      Cranes  NaN       2      yes
```

6. select the rows where the number of visits is less than 4

```
In [21]: df[df['visits']<4]
```

```
Out[21]:         birds  age  visits priority
         0      Cranes  3.5       2      yes
         2     plovers  1.5       3       no
         4  spoonbills  6.0       3       no
         6     plovers  5.5       2       no
         7      Cranes  NaN       2      yes
         8  spoonbills  8.0       3       no
         9  spoonbills  4.0       2       no
```

7. select the rows with columns ['birds', 'visits'] where the age is missing i.e NaN

```
In [112]: df[df['age'].isnull()]
```

```
Out[112]:         birds  age  visits priority
          3  spoonbills  NaN       4        1
          7      Cranes  NaN       2        1
```

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

```
In [193]: df[(df['age']<4) & (df ['birds'] == 'Cranes')]
```

```
Out[193]:     birds  age  visits priority
          0  Cranes  3.5       2      yes
          5  Cranes  3.0       4       no
```

9. Select the rows the age is between 2 and 4(inclusive)

```
In [194]: df[df['age'] <5]
```

```
Out[194]:         birds  age  visits priority
          0      Cranes  3.5       2      yes
          1      Cranes  4.0       4      yes
          2     plovers  1.5       3       no
          5      Cranes  3.0       4       no
          9  spoonbills  4.0       2       no
```

10. Find the total number of visits of the bird Cranes

```
In [32]: df1 = df.groupby('birds')
         df1
```

```
Out[32]: <pandas.core.groupby.groupby.DataFrameGroupBy object at 0x000001A755A18EB8>
```

```
In [50]: df1 = df.groupby('birds')
         for birds, birds_1 in df1:
             print (birds)
             print(birds_1)
         print ('total number of birds visits:' , df1['visits'].sum())
```

```
Cranes
    birds  age  visits priority
0  Cranes  3.5       2      yes
1  Cranes  4.0       4      yes
5  Cranes  3.0       4       no
7  Cranes  NaN       2      yes
plovers
     birds  age  visits priority
2  plovers  1.5       3       no
6  plovers  5.5       2       no
spoonbills
        birds  age  visits priority
3  spoonbills  NaN       4      yes
4  spoonbills  6.0       3       no
8  spoonbills  8.0       3       no
9  spoonbills  4.0       2       no
total number of birds visits: birds
Cranes        12
plovers        5
spoonbills    12
Name: visits, dtype: int64
```

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

```
In [51]: df1['age'].mean()
```

```
Out[51]: 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 [103]: df.loc['10'] = ['plover', 2, 1, 'no']
          df
```

```
Out[103]:          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
5       Cranes  3.0       4        0
6      plovers  5.5       2        0
7       Cranes  NaN       2        1
8   spoonbills  8.0       3        0
9   spoonbills  4.0       2        0
10      plover  2.0       1       no
```

```
In [106]: df.drop(df.index[10])
```

```
Out[106]:         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
          5      Cranes  3.0       4        0
          6     plovers  5.5       2        0
          7      Cranes  NaN       2        1
          8  spoonbills  8.0       3        0
          9  spoonbills  4.0       2        0
```

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

```
In [121]: df.groupby('birds').size()
```

```
Out[121]: birds
          Cranes        4
          plovers       2
          spoonbills    4
          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 [123]: df.sort_values(by='age', ascending=False)
```

```
Out[123]:         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 [62]: df.sort_values(by='visits')

Out[62]:        birds  age  visits priority
         0     Cranes  3.5       2      yes
         6    plovers  5.5       2       no
         7     Cranes  NaN       2      yes
         9 spoonbills  4.0       2       no
         2    plovers  1.5       3       no
         4 spoonbills  6.0       3       no
         8 spoonbills  8.0       3       no
         1     Cranes  4.0       4      yes
         3 spoonbills  NaN       4      yes
         5     Cranes  3.0       4       no
```

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

```
In [126]: df['priority'].replace(['yes' == 1, 'no' == 0])
          df

Out[126]:        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
```

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

```
In [124]: #df.replace([df['birds'] == 'Cranes'] == 'trump')
          df.replace('Cranes', 'trumpeters')

Out[124]:        birds  age  visits priority
          0 trumpeters  3.5       2      yes
          1 trumpeters  4.0       4      yes
          2    plovers  1.5       3       no
          3 spoonbills  NaN       4      yes
          4 spoonbills  6.0       3       no
          5 trumpeters  3.0       4       no
          6    plovers  5.5       2       no
          7 trumpeters  NaN       2      yes
          8 spoonbills  8.0       3       no
          9 spoonbills  4.0       2       no
```