# pandas_basics_practice

March 29, 2020

**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.**

[24]:
```
import pandas as pd
import numpy as np
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
```

[24]:
```
        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.**

[25]:
```
birds.describe()
```

```
[25]:              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
```

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

```
[26]: birds[0:2]
```

```
[26]:     birds  age  visits priority
       a  Cranes  3.5       2      yes
       b  Cranes  4.0       4      yes
```

```
[27]: birds.head(2)
```

```
[27]:     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**

```
[28]: birds[['birds','age']]
```

```
[28]:         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
```

```
[29]: lst=['birds','age']
       birds[lst]
```

```
[29]:         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’]**

```
[30]: lst=[2,3,7]
      col=['birds','age','visits']
      birds[col].iloc[lst]
```

```
[30]:         birds  age  visits
      c     plovers  1.5       3
      d  spoonbills  NaN       4
      h      Cranes  NaN       2
```

```
[31]: birds.iloc[lst][col]
```

```
[31]:         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**

```
[32]: birds[birds.visits < 4]
```

```
[32]:         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
```

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

```
[33]:         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**

```
[34]: birds[['birds','visits']][birds['age'].isna()]
```

```
[34]:         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**

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

```
/home/manojyamasani/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

```
[35]:     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)**

```
[36]: birds[(birds['age'] >=2) & (birds['age']<=4)]
```

```
[36]:         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**

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

```
[37]: 12
```

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

```
[38]: g=birds.groupby('birds')
      g['age'].mean()
```

```
[38]: 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.**

[39]:
```
k=pd.DataFrame({'birds':['knightangle'],
                'age': [4.5],
                'visits':[5],
                'priority': ['yes']},index=['k'])
birds=birds.append(k)
birds
```

[39]:
```
         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  knightangle  4.5       5      yes
```

[40]:
```
birds=birds.drop('k')
birds
```

[40]:
```
        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)**

[41]:
```
g=birds.groupby('birds')
g['birds'].count()
```

[41]:
```
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.**

[42]:
```
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
```

[42]:
```
        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
```

[43]:
```
birds=birds.sort_values(by=['age','visits'], ascending=[False,True])
birds
```

[43]:
```
        birds  age  visits priority
i  spoonbills  8.0       3       no
e  spoonbills  6.0       3       no
g     plovers  5.5       2       no
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

[44]:
```
birds['priority'] = birds['priority'].map({'yes': 1, 'no': 0})
birds
```

[44]:
```
        birds  age  visits  priority
i  spoonbills  8.0       3         0
e  spoonbills  6.0       3         0
```

```
g     plovers  5.5       2         0
j  spoonbills  4.0       2         0
b      Cranes  4.0       4         1
a      Cranes  3.5       2         1
f      Cranes  3.0       4         0
c     plovers  1.5       3         0
h      Cranes  NaN       2         1
d  spoonbills  NaN       4         1
```

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

[45]:
```
for i in range(len(birds)):
    if birds['birds'].iloc[i]=='Cranes':
        birds['birds'].iloc[i]='trumpeters'
birds
```

[45]:
```
        birds  age  visits  priority
i  spoonbills  8.0       3         0
e  spoonbills  6.0       3         0
g     plovers  5.5       2         0
j  spoonbills  4.0       2         0
b  trumpeters  4.0       4         1
a  trumpeters  3.5       2         1
f  trumpeters  3.0       4         0
c     plovers  1.5       3         0
h  trumpeters  NaN       2         1
d  spoonbills  NaN       4         1
```