# pandas_basics_practice

December 29, 2023

```
[1]: #**Consider the following Python dictionary data and Python list labels:**

     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']
```

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

```
[2]: bird = pd.DataFrame(data, index=labels)
```

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

```
[3]: bird.describe()
```

```
[3]:             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**

```
[4]: bird.head(2)
```

```
[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**

```
[5]: print(bird[['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’]**

```
[6]: bird[['birds', 'age', 'visits']].loc[['b', 'c', 'g']]
```

```
[6]:      birds  age  visits
     b   Cranes  4.0       4
     c  plovers  1.5       3
     g  plovers  5.5       2
```

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

```
[7]: bird[bird['visits']<4]
```

```
[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**

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

```
[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**

```
[9]: bird[(bird['birds'] =='Cranes') & (bird['age']<4)]
```

```
[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)**

```
[10]: bird[(bird['age']>=2) & (bird['age']<=4)]
```

```
[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**

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

```
[11]: 12
```

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

```
[12]: # Without using groupby
      for bird_name in bird['birds'].unique():
          print(bird[bird['birds'] == bird_name]['age'].mean())
```

```
3.5
3.5
6.0
```

```
[13]: # Using groupby
      bird.groupby('birds')['age'].mean()
```

```
[13]: 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.**

```
[14]: mydf = pd.DataFrame([['parrot', 2, 3, 'yes']], columns=bird.columns,␣
       ↪index=['k'])
```

```
[15]: mydf
```

[15]:
```
    birds  age  visits priority
k  parrot    2       3      yes
```

[16]:
```
bird = pd.concat([bird, mydf])
```

[17]:
```
bird.drop(labels='k', inplace=True)
```

[18]:
```
bird
```

[18]:
```
        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)

[19]:
```
# 1st method
for bird_name in bird['birds'].unique():
    print(bird_name, bird[bird['birds'] == bird_name]['birds'].count())
```

```
Cranes 4
plovers 2
spoonbills 4
```

[20]:
```
# Using groupby
bird.groupby('birds')['birds'].count()
```

[20]:
```
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.

[21]:
```
bird.sort_values('age', ascending=False).sort_values('visits')
```

[21]:
```
        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

[22]:

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

[23]:

```
bird
```

[23]:

```
        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’.

[24]:

```
for row in bird.iterrows():
    bird_name = row[1]['birds']
    if bird_name == 'Cranes':
        row[1]['birds'] = 'trumpeters'
    bird.loc[row[0]] = row[1]
```

[25]:

```
# Using builtin function of pandas
bird['birds'].replace('trumpeters', 'Cranes', inplace=True)
```

[26]:

```
bird
```

[26]:

```
        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
```

[ ]:

```

```