

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', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

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

```
In [70]: 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', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df=pd.DataFrame(data,index=labels)
df
```

Out[70]:

| | 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 [71]: df.describe()

Out[71]:
```

| | 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 [72]: df.iloc[0:2,:]
# df.head(2)

Out[72]:
```

| | 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 [73]: df[['birds', 'age']]

Out[73]:
```

| | 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 [74]: df.loc[['b', 'c', 'g'], ['birds', 'age', 'visits']]

Out[74]:
```

| | 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

```
In [75]: df[df.visits<4]

Out[75]:
```

| | 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 [76]: df[df.age.isnull()] [['birds', 'visits']]

Out[76]:
```

| | 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

```
In [77]: df[(df.birds=='Cranes') & (df.age<4)]

Out[77]:
```

| | 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 [78]: df[(df.age>=2) & (df.age<=4)]

Out[78]:
```

| | 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 [79]: df.reset_index(inplace=True) # Resets the index, makes factor a column
df

Out[79]:
```

| | index | birds | age | visits | priority |
|---|-------|------------|-----|--------|----------|
| 0 | a | Cranes | 3.5 | 2 | yes |
| 1 | b | Cranes | 4.0 | 4 | yes |
| 2 | c | plovers | 1.5 | 3 | no |
| 3 | d | spoonbills | NaN | 4 | yes |
| 4 | e | spoonbills | 6.0 | 3 | no |
| 5 | f | Cranes | 3.0 | 4 | no |
| 6 | g | plovers | 5.5 | 2 | no |
| 7 | h | Cranes | NaN | 2 | yes |
| 8 | i | spoonbills | 8.0 | 3 | no |
| 9 | j | spoonbills | 4.0 | 2 | no |

```
In [80]: boolean=[]
for i in df.birds:
    if i=='Cranes':
        boolean.append(True)
    else:
        boolean.append(False)
is_long=pd.Series(boolean)
df[is_long]
print(df['visits'].sum())

29
```

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

```
In [52]: df=pd.DataFrame(data,labels)
df.groupby('birds')['age'].mean()

Out[52]:
```

| | |
|---------------------------|-----|
| 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 [69]: a_row=pd.Series(data=['eagle', '3.2', 5, 'yes'], index=['birds', 'age', 'visits', 'priority'])
#print(k)
row=pd.DataFrame([a_row])
print(type(a_row))
df1=pd.concat([row, df], ignore_index=True)
print(df1)
print(type(df1))
#df=df.concat([df, df1], ignore_index=True)
df1.drop([0,], axis=0, inplace=True)
print(df1)

#print(df)

<class 'pandas.core.series.Series'>
birds age visits priority
0 eagle 3.2 5 yes
1 Cranes 3.5 2 yes
2 Cranes 4 4 yes
3 plovers 1.5 3 no
4 spoonbills NaN 4 yes
5 spoonbills 6 3 no
6 Cranes 3 4 no
7 plovers 5.5 2 no
8 Cranes NaN 2 yes
9 spoonbills 8 3 no
10 spoonbills 4 2 no
<class 'pandas.core.frame.DataFrame'>
birds age visits priority
1 Cranes 3.5 2 yes
2 Cranes 4 4 yes
3 plovers 1.5 3 no
4 spoonbills NaN 4 yes
5 spoonbills 6 3 no
6 Cranes 3 4 no
7 plovers 5.5 2 no
8 Cranes NaN 2 yes
9 spoonbills 8 3 no
10 spoonbills 4 2 no
```

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

```
In [13]: df.groupby('birds').count()

Out[13]:
```

| | 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.

```
In [14]: df.sort_values(by=['age', 'visits'], ascending=[False, True])
#result_sort=df.sort_values(by=['name', 'score'], ascending=[True, True])

Out[14]:
```

| | 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

```
In [15]: df['priority']=df['priority'].map({'yes':1, 'no':0})
df

Out[15]:
```

| | 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'.

```
In [16]: df['birds']=df.birds.map(lambda x: 'trumpeters' if x=='Cranes' else x)
df

Out[16]:
```

| | 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 |