

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.

#Help from <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

```
In [20]: import pandas as pd
import numpy as np

#data in Dict
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', 'yes', 'no', 'no', 'no']}

#creating dataframe name as birds with index label
birds = pd.DataFrame(data, index=(['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']))
```

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

```
In [21]: birds.head() # To print
```

```
Out[21]:
```

| | 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 |

```
In [22]: birds.tail()
```

```
Out[22]:
```

| | birds | age | visits | priority |
|---|------------|-----|--------|----------|
| 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 |

```
In [23]: birds.describe() #It Calculates all the static values
```

```
Out[23]:
```

| | 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 |

```
In [24]: birds.shape #Total no of rows and columns
```

```
Out[24]: (10, 4)
```

```
In [25]: birds.columns #Column names in table
```

```
Out[25]: Index(['birds', 'age', 'visits', 'priority'], dtype='object')
```

```
In [26]: #Count the values group wise in a dataset
# It ia an Imbalanced dataset
birds["birds"].value_counts()
```

```
Out[26]: spoonbills    4
Cranes                4
plovers               2
Name: birds, dtype: int64
```

3. Print the first 2 rows of the birds dataframe

```
In [27]: birds[:2] # Or else we can use iloc[]
```

```
Out[27]:
```

| | birds | age | visits | priority |
|---|--------|-----|--------|----------|
| a | Cranes | 3.5 | 2 | yes |
| b | Cranes | 4.0 | 4 | yes |

```
In [28]: birds.iloc[:2] # It print using row value
```

```
Out[28]:
```

| | 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 [29]: birds[["birds", "age"]]
```

```
Out[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']

```
In [30]: birds[["birds", "age", "visits"]].iloc[[2,3,7]]
```

```
Out[30]:
```

| | 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

```
In [31]: birds[birds.visits<4]
```

```
Out[31]:
```

| | 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 NaNz

```
In [32]: # taken help from Geeksforgeeks
col=birds[birds['age'].isnull()]
col[["birds", "visits"]]
```

```
Out[32]:
```

| | 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 [33]: birds[(birds["birds"]=="Cranes") & (birds["age"] < 4)]
```

```
Out[33]:
```

| | 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 [34]: birds[(birds["age"]>=2) & (birds["age"]<=4)]
```

```
Out[34]:
```

| | 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 [35]: print("Total no of visits of the bird Cranes =",birds.visits[birds["birds"]=="Cranes"].sum
())

Total no of visits of the bird Cranes = 12
```

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

```
In [36]: print("The Mean Age of bird (Cranes) is ",birds.age[birds["birds"]=="Cranes"].mean())
print("The Mean Age of bird (Plovers) is ",birds.age[birds["birds"]=="plovers"].mean())
print("The Mean Age of bird (Spoonbills) is ",birds.age[birds["birds"]=="spoonbills"].mean
())

The Mean Age of bird (Cranes) is  3.5
The Mean Age of bird (Plovers) is  3.5
The Mean Age of bird (Spoonbills) is  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.

```
In [37]: #Taken help from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFram
e.loc.html
# GeeksforGeeks
# Appending new row

birds.loc['k'] = {'birds': 'Cranes',
                 'age': 4.5,
                 'visits': 3,
                 'priority': 'yes'}

#Appending Data
birds
```

```
Out[37]:
```

| | 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 | Cranes | 4.5 | 3 | yes |

```
In [38]: birds=birds.drop(['k'])
birds
```

```
Out[38]:
```

| | 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 |

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 [39]: #https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.htm
l
sort=birds.sort_values(by=["age", "visits"],ascending=[False, True])
sort
```

```
Out[39]:
```

| | 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 [40]: birds.replace(["yes", "no"], [1,0])
```

```
Out[40]:
```

| | 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 [41]: birds.replace("Cranes", "trumpeters")
```

```
Out[41]:
```

| | birds | age | visits | priority |
|---|------------|-----|--------|----------|
| a | trumpeters | 3.5 | 2 | yes |
| b | trumpeters | 4.0 | 4 | yes |
| c | plovers | 1.5 | 3 | no |
| d | spoonbills | NaN | 4 | yes |
| e | spoonbills | 6.0 | 3 | no |
| f | trumpeters | 3.0 | 4 | no |
| g | plovers | 5.5 | 2 | no |
| h | trumpeters | NaN | 2 | yes |
| i | spoonbills | 8.0 | 3 | no |
| j | spoonbills | 4.0 | 2 | no |

```
In [ ]:
```

```
In [ ]:
```