Assignment - 9 | Pandas

Akash Duttachowdhury

21052386

[Q1 – 16] 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 [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
data = {
    'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes', 'spoonbi
    '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.

In [ ]:
```
df = pd.DataFrame(data)
print(df)
```

```
       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
```

In [ ]:
```
df.index = labels
print(df)
```

```
       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 [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   birds     10 non-null     object 
 1   age       8 non-null      float64
 2   visits    10 non-null     int64  
 3   priority  10 non-null     object 
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

3. Print the first 2 rows of the birds dataframe

In [ ]:
```
df.head(2)
```

Out[ ]:

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

Out[ ]:

	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 [ ]:
```
df.loc[['c', 'd', 'h'], ['birds', 'age', 'visits']]
```

Out[ ]:

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

Out[ ]:

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

Out[ ]:

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

Out[ ]:

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

Out[ ]:

	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

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

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

Out[ ]:
```
12
```

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

In [ ]:
```
df.groupby('birds')['age'].mean()
```

Out[ ]:
```
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 [ ]:
```
df.loc['k'] = ['dodo', 1, 2, 'no']
df
```

Out[ ]:

	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	dodo	1.0	2	no

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

Out[ ]:

	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)

In [ ]:
```
df['birds'].value_counts()
```

Out[ ]:
```
birds
Cranes        4
spoonbills    4
plovers       2
Name: count, 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 [ ]:
```
df.sort_values(by=['age', 'visits'], ascending=[False, True])
```

Out[ ]:

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

Out[ ]:

	birds	age	visits	priority
a	Cranes	3.5	2	NaN
b	Cranes	4.0	4	NaN
c	plovers	1.5	3	NaN
d	spoonbills	NaN	4	NaN
e	spoonbills	6.0	3	NaN
f	Cranes	3.0	4	NaN
g	plovers	5.5	2	NaN
h	Cranes	NaN	2	NaN
i	spoonbills	8.0	3	NaN
j	spoonbills	4.0	2	NaN

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

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

Out[ ]:

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