

Assignment on Pandas Basic

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.

```
In [0]: import numpy as np
from numpy import nan
import pandas as pd
```

```
In [0]: data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cra
'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', 'r
```

```
In [0]: df=pd.DataFrame(data)
```

```
In [14]: 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 [0]: df['labels'] = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

In [16]: `print(df)`

```

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

```

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

In [61]: `df.describe()`

Out[61]:

	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 [62]: `df.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 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
4   labels      10 non-null    object
dtypes: float64(1), int64(1), object(3)
memory usage: 528.0+ bytes

```

3. Print the first 2 rows of the birds dataframe

```
In [25]: df.head(2)
```

```
Out[25]:
```

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b

4. Print all the rows with only 'birds' and 'age' columns from the dataframe

```
In [32]: df[['birds', 'age']]
```

```
Out[32]:
```

	birds	age
0	Cranes	3.5
1	Cranes	4.0
2	plovers	1.5
3	spoonbills	NaN
4	spoonbills	6.0
5	Cranes	3.0
6	plovers	5.5
7	Cranes	NaN
8	spoonbills	8.0
9	spoonbills	4.0

5. select [2, 3, 7] rows and in columns ['birds', 'age', 'visits']

```
In [36]: #https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-location  
df[['birds', 'age', 'visits']].iloc[[2,3,7]]
```

```
Out[36]:
```

	birds	age	visits
2	plovers	1.5	3
3	spoonbills	NaN	4
7	Cranes	NaN	2

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

```
In [38]: df.loc[df['visits']<4]
```

```
Out[38]:
```

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
2	plovers	1.5	3	no	c
4	spoonbills	6.0	3	no	e
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j

7. select the rows with columns ['birds', 'visits'] where the age is missing i.e NaN

```
In [0]: #df
```

```
In [58]: df[['birds', 'visits']].loc[df['age'].isnull()]
```

```
Out[58]:
```

	birds	visits
3	spoonbills	4
7	Cranes	2

```
In [0]: #df.loc[df['age'].isnull()]
```

8. Select the rows where the birds is a Cranes and the age is less than 4

```
In [69]: df[df['birds']=='Cranes'].loc[df['age']<4]
```

```
Out[69]:
```

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
5	Cranes	3.0	4	no	f

9. Select the rows the age is between 2 and 4(inclusive)

```
In [74]: df.loc[(df['age']>2) & (df['age']<=4)]
```

```
Out[74]:
```

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
5	Cranes	3.0	4	no	f
9	spoonbills	4.0	2	no	j

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

```
In [77]: df['visits'].sum()
```

```
Out[77]: 29
```

```
In [90]: (df['visits'].loc[df['birds']=='Cranes']).sum()
```

```
Out[90]: 12
```

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

```
In [95]: df['birds'].unique()
```

```
Out[95]: array(['Cranes', 'plovers', 'spoonbills'], dtype=object)
```

```
In [97]: print("Mean Age of Bird Cranes=", (df['age'].loc[df['birds']=='Cranes']).mean())
print("Mean Age of Bird plovers=", (df['age'].loc[df['birds']=='plovers']).mean())
print("Mean Age of Bird spoonbills=", (df['age'].loc[df['birds']=='spoonbills']).n
```

```
Mean Age of Bird Cranes= 3.5
```

```
Mean Age of Bird plovers= 3.5
```

```
Mean Age of Bird spoonbills= 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 [98]: df
```

```
Out[98]:
```

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

```
In [0]: k = pd.Series({'birds':'crow','age':5,'visits':10,'priority':'yes','labels':'c'},
```

```
In [0]: df = df.append(k)
```

In [116]: df

Out[116]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	yes	a
1	Cranes	4.0	4	yes	b
2	plovers	1.5	3	no	c
3	spoonbills	NaN	4	yes	d
4	spoonbills	6.0	3	no	e
5	Cranes	3.0	4	no	f
6	plovers	5.5	2	no	g
7	Cranes	NaN	2	yes	h
8	spoonbills	8.0	3	no	i
9	spoonbills	4.0	2	no	j
k	crow	5.0	10	yes	c

In [0]: df=df.drop(df.index[10])

In [119]: df

Out[119]:

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

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

```
In [132]: print("number of Cranes of birds in dataframe=", (df['birds']=='Cranes').sum())

print("number of plovers of birds in dataframe=", (df['birds']=='plovers').sum())

print("number of spoonbills of birds in dataframe=", (df['birds']=='spoonbills').sum())
```

number of Cranes of birds in dataframe= 4
number of plovers of birds in dataframe= 2
number of spoonbills of birds in dataframe= 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 [0]: df_age_decending = df.sort_values(by=['age'], ascending=False)
```

```
In [134]: df_age_decending
```

```
Out[134]:
```

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

```
In [0]: df_visits_ascending = df_age_decending.sort_values(by=['visits'], ascending=True)
```

In [136]: `df_visits_ascending`

Out[136]:

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

15. Replace the priority column values with 'yes' should be 1 and 'no' should be 0

In [0]: [#https://stackoverflow.com/questions/40901770/is-there-a-simple-way-to-change-a-c](https://stackoverflow.com/questions/40901770/is-there-a-simple-way-to-change-a-c)
`df['priority'] = df['priority'].map({'yes': 1, 'no': 0})`

In [138]: `df`

Out[138]:

	birds	age	visits	priority	labels
0	Cranes	3.5	2	1	a
1	Cranes	4.0	4	1	b
2	plovers	1.5	3	0	c
3	spoonbills	NaN	4	1	d
4	spoonbills	6.0	3	0	e
5	Cranes	3.0	4	0	f
6	plovers	5.5	2	0	g
7	Cranes	NaN	2	1	h
8	spoonbills	8.0	3	0	i
9	spoonbills	4.0	2	0	j

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

In [0]: [#https://www.geeksforgeeks.org/python-pandas-dataframe-replace/](https://www.geeksforgeeks.org/python-pandas-dataframe-replace/)
`df=df.replace(to_replace ="Cranes", value ='trumpeters')`

In [141]: df

Out[141]:

	birds	age	visits	priority	labels
0	trumpeters	3.5	2	1	a
1	trumpeters	4.0	4	1	b
2	plovers	1.5	3	0	c
3	spoonbills	NaN	4	1	d
4	spoonbills	6.0	3	0	e
5	trumpeters	3.0	4	0	f
6	plovers	5.5	2	0	g
7	trumpeters	NaN	2	1	h
8	spoonbills	8.0	3	0	i
9	spoonbills	4.0	2	0	j