Pandas

Reference link : https://www.kaggle.com/residentmario/creating-reading-and-writing#Getting-started (https://www.kaggle.com/residentmario/creating-reading-and-writing#Getting-started)

In [ ]:

```
# two core objects in pandas: the DataFrame and the Series
```

DataFrames

In [14]:

```
import pandas as pd

# create using dictionary format
# assigns values to the column labels, but just uses an ascending count from 0 (0, 1, 2, 3, ...) for the row labels.
a = pd.DataFrame({ 'Departments': ['CSE', 'ISE', 'ECE', 'EEE', 'BT', 'MATHEMATICS'],
    'email id': ['cse@bmsce.ac.in','ise@bmsce.ac.in','ece@bmsce.ac.in','eee@bmsce.ac.in',
                 'biotech@bmsce.ac.in','math@bmsce.ac.in']})
a

# rows can be named using the 'index' parameter, as the row names are called indexes
# after naming rows
a.index = range(1,7)
a
# change col names
a = a.rename(columns = {'email id': 'Mail ID'})
# Equivalent => a.columns.values[1] = 'Mail ID'
a
```

Out[14]:

	Departments	Mail ID
1	CSE	cse@bmsce.ac.in
2	ISE	ise@bmsce.ac.in
3	ECE	ece@bmsce.ac.in
4	EEE	eee@bmsce.ac.in
5	BT	biotech@bmsce.ac.in
6	MATHEMATICS	math@bmsce.ac.in

SERIES

In [17]:

```
# A Series is, in essence, a single column of a DataFrame.
# So you can assign column values to the Series the same way as before, using an index parameter.
# However, a Series does not have a column name, it only has one overall name

import pandas as pd
b = pd.Series([10,20,30,25,40,50,30],index=['Sun', 'Mon','tues','Wed','Thu','Fri','sat'])
b.name = 'Temp stats (in degree celcius)'
b
```

Out[17]:

```
Sun     10
Mon     20
tues    30
Wed     25
Thu     40
Fri     50
sat     30
Name: Temp stats (in degree celcius), dtype: int64
```

In [18]:

```
# READING DATA
```

In [37]:

```
import pandas as pd
s = pd.read_csv("sample.csv")
print(s.A)
print('------')
s['A'][0]
s['E'] = [10,11]
s
```

```
0    1
1    5
Name: A, dtype: int64
------
```

Out[37]:

	Sl. No.	A	B	C	D	E
0	1	1	2	3	4	10
1	2	5	6	7	8	11

INDEXING

In [51]:

```
# 1.  index-based selection - using iloc
# Both loc and iloc are row-first, column-second ->  it's marginally easier to retrieve ro
ws

import pandas as pd
s = pd.read_csv("sample.csv",)
print(s.iloc[0]) # to access first row
# col can be retrieved using iloc
s.iloc[[0,2],[1,3]]
```

```
A    1
B    2
C    3
D    4
Name: 0, dtype: int64
```

Out[51]:

	B	D
0	2	4
2	12	14

In [57]:

```
# 2.  label-based selection - using loc
# Both loc and iloc are row-first, column-second ->  it's marginally easier to retrieve ro
ws

import pandas as pd
s = pd.read_csv("sample.csv",)
s.loc[1,['B','C']]
```

Out[57]:

```
B    6
C    7
Name: 1, dtype: int64
```

In [58]:

```
# Conditional Selection
```

In [63]:

```
import pandas as pd
s = pd.read_csv("sample.csv",)

print(s[(s.C==3) & (s.D ==14)])
```

```
    A   B  C   D
2  11  12  3  14
```

More

https://www.kaggle.com/residentmario/summary-functions-and-maps#Introduction (https://www.kaggle.com/residentmario/summary-functions-and-maps#Introduction)

In [ ]: