July 12

① csv - separated by commas (,)
② tsv - " " tab (---)

# Python - tutorial 10

## #Pandas : (Panel Data System, a library).

- file Read Write operation :

```
import pandas as pd          → (csv is type of file)
draft1 = pd.read_csv('address.csv')
draft1 /or draft1.head()     ← heading + top 5 data
```

tabular form:

| | a | b | c | d | message |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | hello |
| 1 | 5 | 6 | 7 | 8 | world |
| 2 | 9 | 10 | 11 | 12 | foo |

← by default first line as heading (row & column)

```
type(draft1)
pandas.core.Frame.DataFrame
```

- head() heading + top-5 data sets
- tail() " + bottom-5 data sets
- head(8) " + top-8 data sets
- tail(18) " + bottom 18 data sets

```
draft2 = pd.read_table('test.tsv')
draft2.head()
```

← heading will always be different.

| | test | test1 | test2 | test3 | test4 |
|---|---|---|---|---|---|
| | sudh | sudh | sudh | sudh | sudh |
| | kumar | kumar | kumar | kumar | kumar |

* if you want to read csv file in tsv format:

```
draft1 = pd.read_csv('test.tsv')
draft1.tail()   (∵ no space) ↓
```

```
        test\t test1\t test2\t test3\t test4
0   sudh\t sudh\t sudh\t sudh\t sudh\t sudh
1   kumar\t kumar\t kumar\t kumar\t kumar\t kumar
```

but if we do like ↓:

```
draft1 = pd.read_csv('test.tsv, sep='\t')
draft1.tail()
```

```
        test  test1  test2  test3  test4
sudh    sudh  sudh   sudh   sudh   sudh
kumar   kumar kumar  kumar  kumar  kumar
```

Similarly, for tsv in csv (∵ sep=',')

* if you don't want to consider first row as header instead it will be part of data
(∵ header = None) → & by default heading is [0,1,2,3,4,5]

* if you want to give your own heading:

```
df = pd.read_csv('ex2.csv',
                 names=['M','A','D','H','U',
                        'P','I'])
df
```

(if there are extra column then NaN)

| | M | A | D | H | V | P | I |
|---|---|---|---|---|---|---|---|
| 0 | a | b | c | d | e | NaN | NaN |
| 1 | f | g | h | i | j | NaN | NaN |
| 2 | k | l | m | n | o | NaN | NaN |

⊛ (Multilevel indexing / hierarical indexing) : (When you want to store data inside data).

```
parsed = pd.read_csv('csv_mindex.csv')
parsed
```

| | key1 | key2 | value1 | value2 |
|---|---|---|---|---|
| 0 | one | a | 1 | 2 |
| 1 | one | b | 3 | 4 |
| 2 | one | c | 5 | 6 |
| 3 | two | a | 7 | 8 |
| 4 | two | b | 9 | 10 |
| 5 | two | c | 11 | 12 |

(default index)

```
parsed=pd.read_csv('csv_mindex.csv',
                   index_col=['key1','key2'])
```

| key1 | key2 | value1 | value2 |
|---|---|---|---|
| one | a | 1 | 2 |
| | b | 3 | 4 |
| | c | 5 | 6 |
| two | d | 7 | 8 |
| | e | 9 | 10 |
| | f | | |

(∵ key1 & key2 became column ind and default index is removed.)

## ③ xlsx - excel

(*) if you want to read some data & skip some:
(∵ skiprows = [0, 2, 3])

(*) if there is some null data in our data set & you want to know (True/False):

```
result = pd.read_csv('ex5.csv')
result
pd.isnull(result)
```

True/False

| | a | b | c | d | e |
|---|---|---|---|---|---|
| 0 | False | False | False | False | True |
| 1 | False | False | True | False | False |
| 2 | False | False | False | False | False |

null-data

(*) Excel-sheets :-

```
draft3 = pd.read_excel('...xlsx')
```

(by default first sheet of excel)

or

```
draft3 = pd.read_excel('...xlsx', sheet_name='Sheet2')
```

(for particular sheet)

(*) accessing data from any website (in tabular form)
PANDAS-HTML won't be able to access data present in other than tabular form:

```
url = "http://www.....html"
BB_data = pd.read_html(url)
BB_data
[.....] = [dataframe1, df2, ...]
type(BB_data) → list
```

multiple tables present on website are merged in a list

data-frame → collection of rows & columns.
(m x 1), (1 x n) → series

BB_data[0] → First dataframe of list
df = BB_data[0]
df.head(10) → first 10 rows
df['column15 say'] → extracting data from required columns
(type ↓ Series)
df[['col1', 'col2', 'col15']] → multiple-column
(type ↓ list) (∵ pass as list)

* when you want to know all column name:
df.columns (∵ heading)
↳ Index(['M', 'A', 'D', 'H', 'U', 'P', 'I', 'N'])

* data type of columns:
titanic_train.dtypes

| | |
|---|---|
| passangerId | int64 |
| Name | object (/string) |
| Age | float64 |
| Sex | object |

* Some Questions:
1. read data in df format
2. print first five & last two rows
3. change column name
4. select one column & print its data type
5. print data dtype
6. check if any data is null

* [it is possible to convert list into series & vice-versa.]
* (skiping random rows is possible & not column-wise first we have to read all data & then filter it)

1. df = Pd.read_csv(r'https://....csv', sep=',')
2. df.head()
2. df.tail(2)
3. df.rename(columns={'age': 'Age'}) ← check other way too
4. df['age'].dtype
5. df.dtypes
6. df.isnull().sum()
7. ( , skiprows=[2,3,8])