Pandas

```
import numpy as np
import pandas as pd
In : dict 1 = { "name" : ['harry', 'rohan',
                          'skillf', 'shubh'],
                "marks" : [90, 87, 63, 74],
                "city" : ['rampur', 'delhi', 'kolkata',
                          'Mumbai'] }
In : df = pd.DataFrame(dict1)
```

both D & F are capital

In : df

Out : forms a table to analyse the data in a better way

↓

& also gives index to each row

In : df.to_csv('file.csv') → to import df to spreadsheet

In : df.to_csv('file.csv', index = False)

↳ to make a spread sheet which does not contain the index

In : df.head(2)
In : df.tail(2)

} no. of rows to be seen

In : df.describe()

Output : find mean, σ, σ^2, count etc of all the columns containing numerical value.

In: harry = pd.read_csv('name.csv')

↳ to read a spread sheet

In: harry ['Column name']

↳ prints the particular column

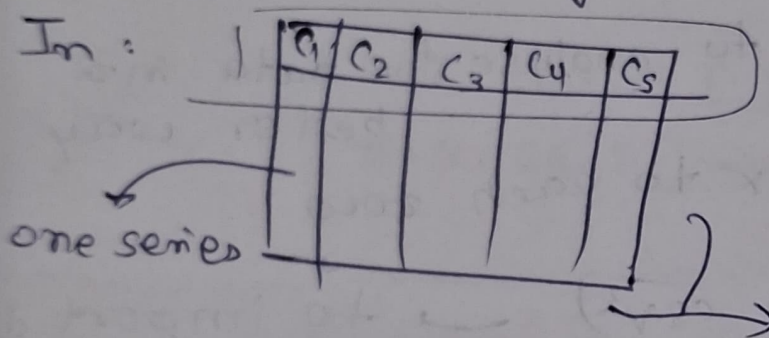
In: harry ['column name'][0] = 50

↳ change the value

In: harry.index = ['a', 'b', 'c', 'd']

↳ to change the indexing

In:



a series consist of row/column containing same element

In: ser = pd.Series(np.random.rand(3))

In: ser ↳ outputs a series from (0,1,2)

In: type(ser) ↳ [Series]

In: newdf = pd.DataFrame(np.random.rand(334,5), index=np.arange(334))

334 rows & 5 col

↳ 2D array (row × column)

↳ creates a new data frame.

In: newdf.index → prints index (0-333)

In: newdf.column → prints column (0-4)

In: newdf.to_numpy()
↳ converts the table to a numpy array

In: newdf.T
↳ transpose of the table (index ⇌ column)

In: newdf.sort_index(axis=0, ascending=False)
↳ upside-down of the table
(ascending=True, by default)

In: newdf.sort_index(axis=1, ascending=False)
↳ lateral inversion of the table

In: newdf[0]
↳ series
prints the column 0

In: newdf2 = newdf
(newdf2 is basically a view of newdf

In: newdf2[0][0] = 97.83

In: newdf → changes

newdf2 is pointing towards newdf
so when we change newdf2, newdf also changes

In : newdf2 = newdf[:] or newdf.copy()

↳ now on changing newdf2, newdf will not change

In : newdf.loc[0,0] = 654

↳ correct way to change any element

In : newdf.column = list["ABCDE"]

In : newdf.drop(0, axis=1)

↳ remove from column = 0

In : newdf.loc[[1,2],['C','D']]

↳ will give only a part of the table

(if newdf = newdf.loc[[1,2],['C','D']]

↳ then newdf would have changed to this shorter table)

(if we write or inplace=True)

In : newdf.loc[:,['C','D']]

↳ prints entire 'C & D' column

In : newdf.loc[[1,2],:]

↳ prints entire row 1 & 2

In : newdf.loc[(newdf['A']<0.3)] && condition

In : newdf.iloc[0,4] ← only indexing

In : newdf['B'].isnull() && newdf['B'].notnull() ← bool

In : newdf.loc['B'] = None ← changes to None

In : newdf.info()

In : newdf.describe()

In : newdf['A'].value_counts()