DATA SCIENCE(study of data)

The process of using data to find solution for a problem
or
The process of using data to predict the outcome of a problem

library or package:set of pre defined,related functions

eg:
"math" library has functions related to mathematics

```
>>> import math          #to use any function in "math" library ,firs u have to import it into your program

>>> math.sqrt(81)
9.0
>>> math.floor(81.9)
81.0
>>> math.ceil(81.9)
82.0
>>> math.trunc(81.98)
81
>>> math.cos(0)
1.0
>>> math.pow(2,3)
8.0
>>> math.factorial(5)
120
>>> math.fmod(7,3)
1.0
```

numpy

It is a library(package)(set of predefined functions)for the Python programming language, adding support for multi-dimensional arrays.

NumPy is used for working with arrays.

NumPy is short for "Numerical Python".

to use any function in "numpy",firs u have to import it into your program like

import numpy #it should be the first statement in your program

```
>>> import numpy
>>> a=numpy.array([[1,2,3],[5,6,7],[8,9,10]])
>>> a[0]
array([1, 2, 3])
>>> a[1]
array([5, 6, 7])
>>> a[2]
array([ 8,  9, 10])
>>> a[0][1]
2
>>> a[-1]
array([ 8,  9, 10])
>>> a[-2]
array([5, 6, 7])
>>> a.shape   #it tells you how to visualize array
(3, 3)             #it means 3 rows and 3 columns
>>>
```

arrange() # it generates range of numbers

```
>>> b=numpy.arange(10)
>>> b
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> b.shape
(10,)    #only 10 elements
>>> c=numpy.arange(0,10,2)
>>> c
array([0, 2, 4, 6, 8])
>>> c.shape
(5,)

>>> c=numpy.arange(10,0,-1)
>>> c
array([10,  9,  8,  7,  6,  5,  4,  3,  2,  1])
>>> c=numpy.arange(10,0,-2)
>>> c
array([10,  8,  6,  4,  2])
```

creating 2D array

```
>>> f=numpy.array([[1,2,3],[4,5,6],[7,8,9]])
>>> f
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
>>> f.shape
(3, 3)
>>> f[-1][0]
7
```

creating 3D array

```
>>> d=numpy.array([[[1,2],[3,4]],[[5,6],[7,8]],[[9,10],[11,12]]])
>>> d
array([[[ 1,  2],
        [ 3,  4]],

       [[ 5,  6],
        [ 7,  8]],

       [[ 9, 10],
        [11, 12]]])
>>> d[0]
array([[1, 2],
       [3, 4]])
>>> d.shape
(3, 2, 2)          # 3 planes ,each plane having 2 rows and 2 columns
```

---

reshaping the array

```
>>> x
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
>>> x=x.reshape(4,2,2)
>>> x
array([[[ 0,  1],
        [ 2,  3]],

       [[ 4,  5],
        [ 6,  7]],

       [[ 8,  9],
        [10, 11]],

       [[12, 13],
        [14, 15]]])
>>> x.shape
(4, 2, 2)
```

---
transpose of 3x3 matrix

```
>>> b
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
>>> b=b.reshape(1,3,3)
>>> b
array([[[0, 1, 2],
        [3, 4, 5],
        [6, 7, 8]]])
>>> b.T          # u can use b.transpose() also
array([[[0],
        [3],
        [6]],

        [[1],
        [4],
        [7]],

        [[2],
        [5],
        [8]]])
```

---
arithmetic operations on array

```
>>> import numpy as n
>>> b=n.arange(9)
>>> b
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
>>> b*2
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16])
>>> b+2
array([ 2,  3,  4,  5,  6,  7,  8,  9, 10])
>>> b**2
array([ 0,  1,  4,  9, 16, 25, 36, 49, 64])
>>> b/2
array([0, 0, 1, 1, 2, 2, 3, 3, 4])
>>> b%2
array([0, 1, 0, 1, 0, 1, 0, 1, 0])
>>> b*b
array([ 0,  1,  4,  9, 16, 25, 36, 49, 64])
>>> b+b
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16])
>>> b-b
array([0, 0, 0, 0, 0, 0, 0, 0, 0])
```

sorting

```
f=n.array([2,4,11,1,0,99,22])
>>> f.sort()
>>> f
array([ 0,  1,  2,  4, 11, 22, 99])
>>> f[4]
11
```

```
>>> f.sum()
139
>>> f.min()
0
>>> f.max()
99
>>> f.mean()
19.857142857142858
```

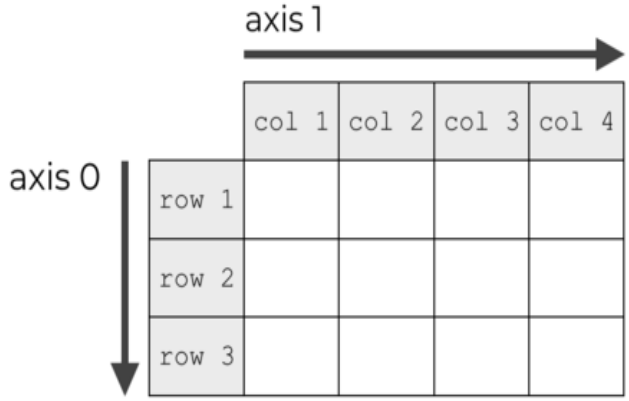
```
>>> import numpy as n
>>> a=n.arange(12)
>>> a
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
>>> a=a.reshape(3,2,2)
>>> a
array([[[ 0,  1],
        [ 2,  3]],

        [[ 4,  5],
        [ 6,  7]],

        [[ 8,  9],
        [10, 11]]])
>>> a.flatten()
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

---

axes in array




```
>>> s=n.array([[2,33,1],[99,3,11],[7,21,0]])
>>> s
array([[ 2, 33,  1],
       [99,  3, 11],
       [ 7, 21,  0]])
>>> s.max()
99
>>> s.max(axis=0)
array([99, 33, 11])
>>> s.max(axis=1)
array([33, 99, 21])
>>> s.min(axis=0)
array([2, 3, 0])
>>> s.min(axis=1)
array([1, 3, 0])
```

```
a=n.array([[3,4,5],[3,4,5]])
>>> b=n.array([[3,4,5],[3,4,5]])
>>> a+b
array([[ 6,  8, 10],
       [ 6,  8, 10]])
>>> a-b
array([[0, 0, 0],
       [0, 0, 0]])
>>> a*b
array([[ 9, 16, 25],
       [ 9, 16, 25]])
>>> a/b
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.]])
>>> a.size
6
```

---

cumulative sum
---
p
array([[1, 2, 3],
[4, 5, 6]])
>>> p.cumsum(axis=1)
array([[1, 3, 6],
[4, 9, 15]])

swapping axes(i.e converting axis 0 to 1 and vice versa)
---
p
array([[1, 2, 3],
[4, 5, 6]])
>>> n.swapaxes(p,0,1)
array([[1, 4],
[2, 5],
[3, 6]])

adding elements horizontally or vertically

>>> p=n.array([[1,2,3],[4,5,6]])
>>> p
array([[1, 2, 3],
[4, 5, 6]])
>>> q=n.array([[11,22,23],[42,52,66]])
>>> q
array([[11, 22, 23],
[42, 52, 66]])
>>> n.hstack((p,q))
array([[1, 2, 3, 11, 22, 23],
[4, 5, 6, 42, 52, 66]])
>>> n.vstack((p,q))
array([[1, 2, 3],
[4, 5, 6],
[11, 22, 23],
[42, 52, 66]])
---
using data types in arrays

>>> a=n.array([0,1,0,1],dtype=int)
>>> a
array([0, 1, 0, 1])

>>> a=n.array([0,1,0,1],dtype=float)
>>> a

```
array([ 0.,  1.,  0.,  1.])

>>> a=n.array([1.3,2.5,3.5,4.6],dtype=int)
>>> a
array([1, 2, 3, 4])

>>> a=n.array([0,1,0,1],dtype=bool)
>>> a
array([False,  True, False,  True], dtype=bool)
>>> p
array([[1, 2, 3],
       [4, 5, 6]])
>>> p<0
array([[False, False, False],
       [False, False, False]], dtype=bool)
>>> p<12
array([[ True,  True,  True],
       [ True,  True,  True]], dtype=bool)
>>> p>4
array([[False, False, False],
       [False,  True,  True]], dtype=bool)
```

---

filling all elements with zeros

```
>>> a=n.zeros(5)
>>> a
array([ 0.,  0.,  0.,  0.,  0.])
>>> a=n.zeros(5,dtype=float)
>>> a
array([ 0.,  0.,  0.,  0.,  0.])
>>> a=n.zeros(5,dtype=int)
>>> a
array([0, 0, 0, 0, 0])
```

```
a=n.array([[3,4,5],[3,4,5]])
>>> b=n.array([[3,4,5],[3,4,5]])
>>> a+b
array([[ 6,  8, 10],
       [ 6,  8, 10]])
>>> a-b
```

```
array([[0, 0, 0],
       [0, 0, 0]])
>>> a.size
6
---------------------------

splitting an array
------------------
>>> a=n.array([1,2,3,4,5,6,7,8,9])
>>> a
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> n.array_split(a,3)
[array([1, 2, 3]), array([4, 5, 6]), array([7, 8, 9])]
>>> y=n.array_split(a,3)
>>> y
[array([1, 2, 3]), array([4, 5, 6]), array([7, 8, 9])]
>>> y=n.array_split(a,4)
>>> y
[array([1, 2, 3]), array([4, 5]), array([6, 7]), array([8, 9])]
>>> y[0]
array([1, 2, 3])
-----------

searching an array
------------------
import numpy
arr = numpy.array([1, 2, 3, 4, 5, 4, 4])
x = numpy.where(arr == 4)
print(x)


import numpy
arr =numpy.array([1, 2, 3, 4, 5, 6, 7, 8])
x = numpy.where(arr%2 == 0)
print(x)

import numpy as n
arr = n.array([2,3,6,9])
x = n.searchsorted(arr, 4)
print(x)

sorting
----------
import numpy as n
arr = n.array([3, 2, 0, 1])
print(n.sort(arr))

import numpy as n
```

```
arr = n.array(['banana', 'cherry', 'apple'])
print(n.sort(arr))

import numpy as n
arr = n.array([[3, 2, 4], [5, 0, 1]])
print(n.sort(arr))

generating random array
----------

from numpy import random
x=random.randint(100, size=(5))
print(x)

-------
from numpy import random
x = random.randint(100, size=(3, 5))
print(x)
--------
import numpy
x = numpy.random.randint(100, size=(3, 5))
print(x)

above 2 codes are same
```

pandas library

Pandas is used to analyze data.

Pandas deals with the following three data structures

Series

Series is a one-dimensional array like data

```
import pandas as p
h=p.Series(['sai','sam','ram'])
print(h)

import pandas as p
data=['sai','sam','ram']
h=p.Series( data)
print(h)
```

```
import pandas as p
data=['sai','sam','ram']
index=[6,7,8]
h=p.Series( data, index)
print(h)

import pandas as p
data=['sai','sam','ram','hari','raj']
index=[6,7,8,9,10]
h=p.Series( data, index)
print(h[0:3])# u can use slicing
```

u can create a series with dictionary also

```
import pandas as p
day= {'s' :'sun', 'm' :'mon', 't' :'tue','w':'wed'}
s = p.Series(day)
print (s)

#here dictionary keys are used as index
```

note:

scalar means single value

a=[2]

so a is scalar

DataFrame(table of contents)

DataFrame is a two-dimensional array of data(a table)
with labels

```
import pandas as p
import numpy as n
```

```
marks=n.array([[33,44,55],[100,90,35],[66,30,99]])
h=p.DataFrame(marks)
print(h)
```

```
import pandas as p
import numpy as n
marks=n.array([[33,44,55],[100,90,35],[66,30,99]])
rlab=['ram','sam','raj']
clab=['telugu','hindi','maths']
h=p.DataFrame(marks,rlab,clab)
print(h)
```

column selection

```
import pandas as p
import numpy as n
marks=n.array([[33,44,55],[100,90,35],[66,30,99]])
rlab=['ram','sam','raj']
clab=['telugu','hindi','maths']
h=p.DataFrame(marks,rlab,clab)
print(h['hindi'])
```

adding columns

```
import pandas as p
import numpy as n
marks=n.array([[33,44,55],[100,90,35],[66,30,99]])
rlab=['ram','sam','raj']
clab=['telugu','hindi','maths']
h=p.DataFrame(marks,rlab,clab)
h['total']=h['telugu']+h['hindi']+h['maths']
print(h)
```

getting data from files(data stored on hard disk)

```
import pandas as p
x=p.read_csv('mydata.csv')# loading from .csv file
print(x)

import pandas as p
x=p.read_excel('h.xls')#loading from .xls file(excel sheet)
print(x)
```

---

finding frequency count

```
import pandas as p
x=p.read_csv('svne.csv')# loading from .csv(comma seperated values)
file
print(x['city'].value_counts())
```

droping duplicate rows

```
import pandas as p
x=p.read_csv('svne.csv')# loading from .csv(comma seperated values)
file
print(x.drop_duplicates())
```

grouping data

```
import pandas as p
x=p.read_csv('svec-marks.csv')# loading from .csv(comma seperated
values) file
print(x.groupby(by='gen').hin.max())
```

checking empty cells

```
import pandas as p
```

```
x=p.read_csv('svec.csv')# loading from .csv(comma seperated values)
file
print(p.isnull(x))
```

filling empty cells

```
------------------
import pandas as p
x=p.read_csv('svec.csv')# loading from .csv(comma seperated values)
file
x['rank'].fillna(-1, inplace=True)
print(x)
```

retrieving a sample row

```
------------------
import pandas as p
x=p.read_csv('svec.csv')# loading from .csv(comma seperated values)
file
print(x.sample())
```

count empty cells in each column

```
------------------
import pandas as p
x=p.read_csv('svec.csv')# loading from .csv(comma seperated values)
file
print(x.isnull().sum())
```

Panel

Panel is a (3D)three-dimensional array of data

```
import pandas as p
import numpy as n
data = {'cse' :n.array([[11,22,23],[42,52,66],[33,44,55]]),
                'eee' :n.array([[1,2,3],[4,5,6],[3,4,5]]),
                'civ' :n.array([[21,22,23],[24,25,26],[33,34,35]])}
```

```
rose=p.Panel(data)
print (rose['cse'])
```

---

displaying data by axes

In panel there are two axes they are

major axis(row wise)

minor axis(column wise)

```
import pandas as p
import numpy as n
data = {'cse' :n.array([[11,22,23],[42,52,66],[33,44,55]]),
                'eee' :n.array([[1,2,3],[4,5,6],[3,4,5]]),
                'civ' :n.array([[21,22,23],[24,25,26],[33,34,35]])}
rose=p.Panel(data)
print (rose.major_xs(0))#all rows at 0 index from all frames
```

```
import pandas as p
import numpy as n
data = {'cse' :n.array([[11,22,23],[42,52,66],[33,44,55]]),
                'eee' :n.array([[1,2,3],[4,5,6],[3,4,5]]),
                'civ' :n.array([[21,22,23],[24,25,26],[33,34,35]])}
rose=p.Panel(data)
print (rose.minor_xs(0))#all columns at 0 index from all frames
```

creating a panel by 3 .csv files

```
import pandas as p
x=p.DataFrame(p.read_csv('svec.csv'))# loading from .csv(comma
seperated values) file
y=p.DataFrame(p.read_csv('svce.csv'))
z=p.DataFrame(p.read_csv('svne.csv'))
data={'svec':x,'svce':y,'svne':z}
rose=p.Panel(data)
```

```
print (rose['svne'])
```

```
import pandas as p
x=p.DataFrame(p.read_csv('svec.csv'))# loading from .csv(comma
seperated values) file
y=p.DataFrame(p.read_csv('svce.csv'))
z=p.DataFrame(p.read_csv('svne.csv'))
data={'svec':x,'svce':y,'svne':z}
rose=p.Panel(data)
print (max(rose['svne'].age))
print (min(rose['svne'].age))
print (sum(rose['svne'].age))
```

note:

series:it is 1D array of data

	eng	hin	math
sam	1	2	3

dataframe:it is array of series

	eng	hin	math
sam	1	2	3
ram	4	5	6
siri	7	8	9

---

panel:it is an array of dataframes

	eng	hin	math
sam	1	2	3
ram	4	5	6
siri	7	8	9
giri	11	22	33
vani	43	53	63
bob	75	85	95
raj	21	22	23
moni	34	53	36
siva	77	78	79

Matplotlib(graph plotting(drawing) library)

Matplotlib is one of the most popular Python
packages used for data
visualization(to create a graphical representation of data)

note: "A picture is worth a thousand words"

ploting lines

```
import matplotlib.pyplot as plt
marks = [80, 85, 90, 25, 20,15,20,99,100]
roll = [10,11,12,13,14,15,16,17,18]
plt.plot(roll,marks)
plt.xlabel("roll number")
plt.ylabel("python marks")
plt.show()
```

marker

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o')
plt.show()

#if we dont specify, defaultly x-axis values ranges
    from 0,1,2,3,...etc
```

plotting multiple lines

```
import matplotlib.pyplot as plt
sub=["tel",'hin','eng','math','sci']
siri= [30,55,95,85,25]
vini =[40,77,99,90,55]
plt.plot(sub,siri,color="red",marker='o')
plt.plot(sub,vini,color="green",marker='o')
plt.show()
```

labels

```
import matplotlib.pyplot as plt
cases =[2,5,30,40,50,50,30,20]
month=['jan','feb','mar','apr','may','jun','jul','aug'])
plt.plot(month, cases)
plt.xlabel("month")
plt.ylabel("cases in thounds")
plt.title("covid outbreak analysis-2021")
plt.show()
```

scatter plot

```
import matplotlib.pyplot as plt
cases =[2,5,30,40,50,50,30,20]
month=['jan','feb','mar','apr','may','jun','jul','aug'])
```

```
plt.scatter(month, cases)
plt.xlabel("month")
plt.ylabel("cases in thounds")
plt.title("covid outbreak analysis-2021")
plt.show()
```

histogram

```
import matplotlib.pyplot as plt
hindi =[30,30,40,40,50,50,60,60,60,60,70,70,70,70]
plt.hist(hindi)
plt.title("hindi marks")
plt.show()
```

pie chart

```
import matplotlib.pyplot as plt
y =[40, 90, 25, 10]
mylabels = ["ece", "eee", "civ", "mec"]
plt.pie(y, labels = mylabels,autopct="%f")
plt.title("result analysis")
plt.show()
```

Multiple subplots in one figure

With the subplot() function you can draw multiple plots in one figure

```
import matplotlib.pyplot as plt

#plot 1:
price =[15,10,40,10,35,35]
month= ['jan','feb','mar','apr','may','june']

plt.subplot(1, 2, 1)
plt.plot(month,price)
plt.title("prices of onion in 2020")
```

```
#plot 2:
price = [30,35,40,40,45,30]
month= ['jan','feb','mar','apr','may','june']

plt.subplot(1, 2,2 )
plt.plot(month,price)
plt.title("prices of onion in 2021")

plt.show()
```

---

ploting on file content

```
import pandas as p
import matplotlib.pyplot as plt
x=p.DataFrame(p.read_csv('svec-marks.csv'))# loading from
.csv(comma seperated values) file
x['tot']=x['hin']+x['math']+x['eng']
x['perc']=x['tot']/300*100
p=list(x['perc'])
s=list(x['name'])
plt.plot(s,p,marker='o',ms=10,c='r')
plt.show()
```