'''

Online Python Compiler.

Code, Compile, Run and Debug python program online.

Write your code in this editor and press "Run" button to execute it.

'''

'''

Online Python Compiler.

Code, Compile, Run and Debug python program online.

Write your code in this editor and press "Run" button to execute it.

'''

'''

Online Python Compiler.

Code, Compile, Run and Debug python program online.

Write your code in this editor and press "Run" button to execute it.

'''

import random as ran

import numpy as np

'''

x = np.array([[ 0, 1, 2],[ 3, 4, 5],[ 6, 7, 8],[ 9, 10, 11]])

print("The array is :",x)

print("Mean along axis 0:",np.mean(x,axis=0),"Standered deviation along axis 0: ",np.std(x, axis=0))

print("Mean along axis 1:",np.mean(x,axis=1),"Standered deviation along axis 1:", np.std(x, axis=1))

'''

'''

l1=np.arange(9).reshape(3,3)

print("Before transpose:",l1)

test=np.transpose(l1)

print("After transpose:",test)

'''

'''

x=np.arange(9).reshape(3,3)

print(x)

y=x.astype(float)

print("After converting to float type:\n",y)

y[1,1]=np.NaN

y[0,0]=np.NaN

y[2,2]=np.NaN

print("Adding Nan value:\n",y)

col\_mean=np.nanmean(y,axis=0)

print("\n The column means are :",col\_mean)

pos=np.where(np.isnan(y))

print("the position where nan available is :",pos)

y[pos]=np.take(col\_mean,pos[1])

print("\nAfter removing nan: ",y)

'''

'''

'diff between ravel() and flatten()'

x=np.arange(10,25,1).reshape(3,5)

print("The original 2d array:",x)

y=x.flatten()

print("After appalying flatten(), original 2d array:",x)

print("After appalying flatten(), resaved 2d array:",y)

z=x.ravel()

print("After appalying ravel(), original 2d array:",x)

print("After appalying ravel(), resaved 2d array:",z)

'''

'''

'how to check weather any dimensional array contains null or nan in a single yes or no reply:'

x=np.arange(5,17,0.5).reshape(4,6)

print("Original array:\n",x)

x[1,3]=np.NaN

x[0,0]=np.NaN

x[2,5]=np.NaN

x[3,2]=np.NaN

print("After nan insertion: \n",x)

print("Is there any nan value:",np.any(np.isnan(x)))

'''

'''

'replace nan with zero'

x=np.arange(5,17,0.5).reshape(4,6)

print(x)

x[1,3]=np.NaN

x[0,0]=np.NaN

x[2,5]=np.NaN

x[3,2]=np.NaN

print("After nan insertion: \n",x)

test=np.nan\_to\_num(x)

print("Replacing nan with zero: ",test)

'''

'''

'to find the max frequency os element in a 1 d array:'

x=np.random.randint(0, 20,10)

print(x)

test=np.bincount(x)

print("Frequency of each digit:",test)

max1=np.argmax(test)

print("Maximum occurance of letter:",max1)

'''

'''

'replace a value on a given condition'

x=np.random.randint(0, 10,20).reshape(4,5)

print(x)

x[x > 6]=100

print("The array after modification: ",x)

'''

'''

'sort a 1 d array in numpy'

x=np.random.randint(0, 10,20)

print(x)

SrtArrindex=np.argsort(x)

print("After sorting the index values: ", SrtArrindex)

print("The sorted array(ascending order: ",x[SrtArrindex])

'''

'sort a 1 d array in numpy using argpartition'

'''

x=np.random.randint(10, 20,10)

print(x)

print("After sorting the index values: ", np.argpartition(x,(1,8)))

print("The sorted array ascending order: ",x[np.argpartition(x,(1,8))])

print(x)

'''

'''

'To find the unique elements in an array containig repetations'

x=np.random.randint(10,20,10)

print("original x:",x)

uniq=np.unique(x)

print("The unique element's in x are:",uniq)

'''

'''

'delete any row or column depending on the axis value'

x=np.random.randint(10,20,20).reshape(4,5)

print("original array: ",x)

print("del column: ",np.delete(x,1,0))

print("Original array after del(): ",x)

'''

'''

'adding a row or column at specified index based on axis value'

x=np.random.randint(10,20,20).reshape(4,5)

print("original array: ",x)

print("adding 5 to 1st column of x: ", np.insert(x,0,5,1))

print("adding 5 to 1st row of x: ", np.insert(x,0,5,0))

'''

'''

'To remove rows containing nan in an array:'

a = np.array([

[1, 0, 0],

[0, np.nan, 0],

[0, 0, 0],

[3, np.nan, np.nan],

[2, 3, 4]

])

print("nan coordinates:",np.isnan(a))

mask = np.any(np.isnan(a), axis=1)

print("mask:",a[~mask])

'''

‘’’

'how to find local maxima in an 1 d array'

from scipy.signal import argrelmax

import numpy as np

x = np.asarray([2, 1, 2, 3, 2, 0, 1, 0])

pos\_max=argrelmax(x)

print("Position of local maxima:",pos\_max)

print("local maxima :",x[pos\_max])

‘’’

'''

'how to find local minima in an 1 d array'

from scipy.signal import argrelmin

import numpy as np

x = np.asarray([2, 1, 2, 3, 2, 0, 1, 0])

pos\_min=argrelmin(x)

print("Position of local minima:",pos\_min)

print("local minima :",x[pos\_min])

'''

‘’’

‘**How to find the index of n'th repetition of an item in an array’**

x=np.random.randint(10,15,10)

print(x)

test=int(input("Enter the item to be searched: "))

rep=int(input("Enter which repitation pos you require: "))

c=0

flag=0

for i in np.nditer(x):

if(i==test and rep>0):

rep=rep-1

flag=1

if(rep==0):

break;

c=c+1

if(rep==0):

print("Item present at loc:",c)

elif(rep>0 and flag==1):

print("item available in list but with no such number of times it is present")

else:

print("No such item found.")

‘’’

'''

'To find the determienent'

x=ran.sample(range(5,10),4)

x=np.reshape(x,(2,2))

print(x)

print("The determinent is : ",np.linalg.det(x))

'''

'''

'To find the trace of matrix'

x=np.random.randint(1,9,16)

x=np.reshape(x,(4,4))

print(x)

trace=np.trace(x)

print("The trace of metrix is: ",trace)

'''

'''

'To find the most smallest elemnt in array'

x=np.random.randint(5,16,20).reshape(5,4)

print(x)

print("The min elemement in lit is: ",np.amin(x))

'''

'''

'whiten() test'

'whiten() performs the operation (numpy 2d array)/(stdeviation colnwise of the array)'

import numpy as np

import scipy.cluster as sc

x=np.random.randint(10,20,16).reshape(4,4)

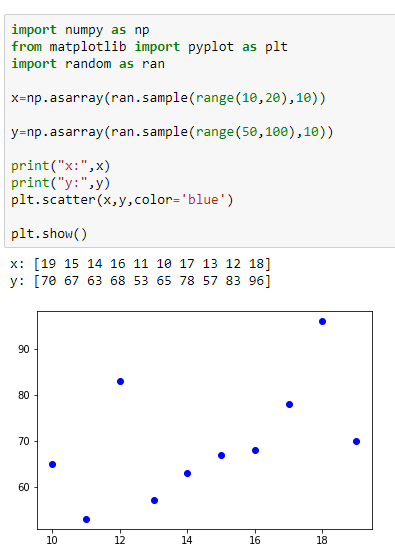
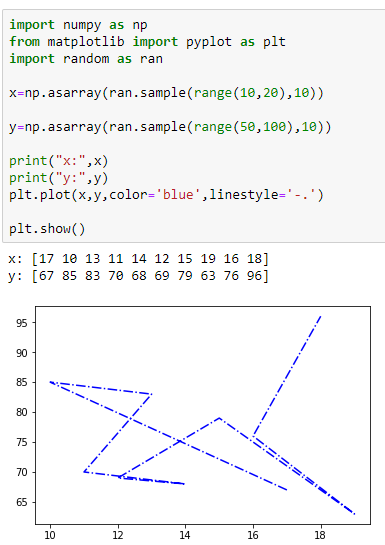
print("original array: ",x)

std\_coln=np.std(x,axis=0)

print("std along axis=0: ",std\_coln)

print("Normalize x : ",sc.vq.whiten(x))

print("Testing whiten with std\_coln: ",(x/std\_coln))

'''

