＃python创建随机数

1. 使用random.random(low,hight) 返回一个位于［low，hight］间的整数，会执行参数检查

>>>import random

>>>random.randint(1,10)

7

1. random.random() 返回一个[0.0 , 1.0]间的浮点数

>>>import random

>>> random.random()

0.5885821552646049

1. random.uniform(val1, val2) 不要求val1小于val2，返回两个数间的一浮点数

>>> random.uniform(1,5.0)

4.485403087612088

>>> random.uniform(9.9, 2)

5.189511116007191

1. random.randrange(start, stop, step) step是步长，三个参数都要求是整数（或小数位为0），当start>stop，step<0

>>> random.randrange(1, 100, 2) #返回[1,100]之间的奇数

19

>>> random.ranrange(100, 1, -2) #返回[100,1]之间的偶数

2

#python生成随机数组

1. 自己编写

import random

def random\_int\_list(start, stop, length):

start, stop = (int(start), int(stop)) if start <= stop else (int(stop), int(start))

length = int(abs(length)) if length else 0

random\_list = []

for i in range(length):

random\_list.append(random.randint(start, stop))

return random\_list

1. np.random.rand 用于生成[0.0, 1.0]之间的随机浮点数， 当没有参数时，返回一个随机浮点数，当有一个参数时，返回该参数长度大小的一维随机浮点数数组，参数建议是整数型

import numpy as np

>>> np.random.rand(10)

1. np.random.randn该函数返回一个样本，具有标准正态分布

import numpy as np

>>> np.random.randn(10)

1. np.random.randint(low[, high, size]) 返回随机的整数，位于半开区间 [low, high]

import numpy as np

>>> np.random.randint(10,size=10)

array([4, 1, 4, 3, 8, 2, 8, 5, 8, 9])

1. random\_integers(low[, high, size]) 返回随机的整数，位于闭区间 [low, high]

import numpy as np

>>> np.random.random\_integers(5)

1. np.random.shuffle(x) 类似洗牌，打乱顺序；np.random.permutation(x)返回一个随机排列

>>> arr = np.arange(10)

>>> np.random.shuffle(arr)

>>> arr

[1 7 5 2 9 4 3 6 0 8]

>>>> np.random.permutation(10)

array([1, 7, 4, 3, 0, 9, 2, 5, 8, 6])

生成在范围内平均分布的随机数组

import numpy as np

raArray=np.random.random(size=1000000)