# 实验3 NumPy科学计算库

## 1. 实验目的

①掌握NumPy科学计算库的语法规则

②能够运用NumPy完成多维数组和矩阵运算

## 2. 实验内容

根据要求，编写程序，运用NumPy完成多维数组和矩阵运算

1. 实验过程

题目一：

生成一个[0,1)之间均匀分布的随机数数组，包含1000个元素， 随机种子为612。接收用户输入一个1-100之间的数字。打印随机数组中所有索引值可以被输入整数整除的数字，并打印序号和索引值。序号从1开始，依次加1。（提示：输出时可以使用制表符"\t"，对齐各列）

|  |
| --- |
| 请输入一个1-100之间的整数：50 序号  索引值  随机数 1    0    0.1434716297030787 2    50    0.3228752619106986 3    100    0.39412407684983874 ...... 20    950    0.4658222822786575 |

程序运行示例：

① 代码

import numpy

random\_list = [0 for \_ in range(1000)]      # 初始化一个大小为1000的0列表

def work2\_q1\_input\_output(*random\_list*=[]):

    # 输入一个整数

    # 打印其整数倍的随机数

    i = *int*(input("请输入一个1-100之间的整数："))

    if i>0 and i<=100:

        j = 0

        num = 0

        for num in range(0,1000):

            if num % i == 0:

                j += 1

                print('%4d  %4d  %.30f' % (j, num, random\_list[num]))

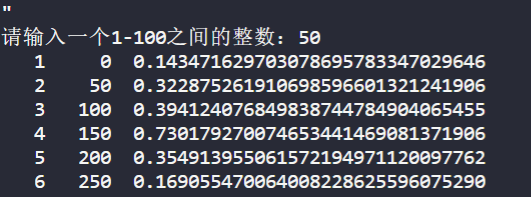
numpy.random.seed(612)      # 设置随机种子

for i in range(0, 1000):    #给列表赋随机值

    random\_list[i] = numpy.random.random()

work2\_q1\_input\_output(random\_list) #调用函数输入整数和打印满足条件数据

② 实验结果



### 题目二：11111

① 代码

import numpy

# 定义数据

x\_list = numpy.array([64.3, 99.6, 145.45, 63.75, 135.46, 92.85, 86.97, 144.76, 59.3, 116.03])

y\_list = numpy.array([62.55, 82.42, 132.62, 73.31, 131.05, 86.57, 85.49, 127.44, 55.25, 104.84])

# 求平均值

x\_mean = numpy.mean(x\_list)

y\_mean = numpy.mean(y\_list)

# 公式求sum和b

w = sum((x\_list - x\_mean)\*(y\_list - y\_mean))/sum((x\_list - x\_mean)\*\*2)

b = y\_mean - w\*x\_mean

# 打印

print(f'w:{w}\nb:{b}')

② 实验结果



### 题目三：

### 22222

① 代码

import numpy

# 定义所需数据列表

x0\_list = [1 for \_ in range(10)]

# x0\_list = numpy.ones()

x1\_list = [64.3, 99.6, 145.45, 63.75, 135.46, 92.85, 86.97, 144.76, 59.3, 116.03]

x2\_list = [2, 3, 4, 2, 3, 4, 2, 4, 1, 3]

y\_list = [62.55, 82.42, 132.62, 73.31, 131.05, 86.57, 85.49, 127.44, 55.25, 104.84]

# 堆叠产生X

# # x\_list = numpy.stack([x0\_list, x1\_list, x2\_list], axis=1)

# matrix array 结果相同

X\_list = numpy.matrix([x0\_list, x1\_list, x2\_list]).transpose()

# reshape转置产生Y

Y\_list = numpy.matrix(y\_list).reshape((10,1))

# 运算

w\_list = (((numpy.linalg.inv(numpy.dot(X\_list.T, X\_list))).dot(X\_list.T)).dot(Y\_list))

# 打印结果

print(f'w\'s shape: {w\_list.shape}')

print(f'X: \n{X\_list}\nY: {Y\_list}\nw: {w\_list}')

# print((numpy.matrix(x0\_list)).shape)

# print(numpy.dot((numpy.matrix(x0\_list).T), (numpy.matrix(x0\_list))))

# print(f'w\'s shape: {w\_list.shape}')

# print(f'X: \n{X\_list}\nY: {Y\_list}\nw: {w\_list}')

# x\_mean = numpy.mean(x\_list)

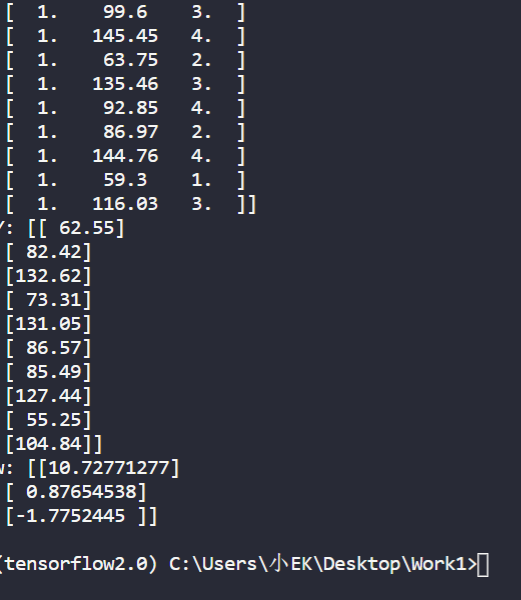
# y\_mean = numpy.mean(y\_list)

# w = sum((x\_list - x\_mean)\*(y\_list - y\_mean))/sum((x\_list - x\_mean)\*\*2)

# b = y\_mean - w\*x\_mean

# print(f'w:{w}\nb:{b}')

② 实验结果



## 实验小结

1. 遇到的问题。

熟悉了numpy的使用。

学会了numpy中matrix和array相关运算和操作，对多维数组有了更深的理解。

② 拓展题(选做)  
请创建一个4×4的随机数组，计算每行每列的最小值，并按照从小到大的顺序输出成4×2的数组。  
# 请创建一个4×4的随机数组，计算每行每列的最小值，并按照从小到大的顺序输出成4×2的数组。  
① 代码  
import numpy

a = numpy.random.randint(0, 100, *size*=(4, 4))  # 随机数组范围为0-100

b = numpy.array([])

print('产生的随机4\*4数组')

print(a)

for i in range(0, 4):  # 每一列最小

    y = a[0, i]

    for j in range(0, 3):

        if y > a[j + 1, i]:

            y = a[j + 1, i]

    b = numpy.append(b, y)

for i in range(0, 4):  # 每一行最小

    x = a[i, 0]

    for j in range(0, 3):

        if x > a[i, j + 1]:

            x = a[i, j + 1]

    b = numpy.append(b, x)

b = numpy.sort(b)  # 从小到大排序

b.resize(4, 2)

print(b)

