第十三周Python实验报告（周三）

**学号**：117060400110 **姓名**：蒙柳双

**班级**：17应用统计学1班 **指导老师**：林卫中老师

**实验目的：**

1.了解并会使用矩阵运算库numpy

2.了解并运用绘制库matplotlib

3.对matplotlib的子库matplotlib.pyplot的熟悉与使用其绘制图

4.会使用以上两个库进行手绘风格图片的使用、绘制科学坐标系和绘制雷达图等等

**实验要求：**

1. 上课不玩手机
2. 要认真编码并且思考
3. 实在不会的就向老师求助

**实验内容步骤：**

1. **解方程**Ax = b

核心代码：

import numpy as np

from numpy.linalg import inv

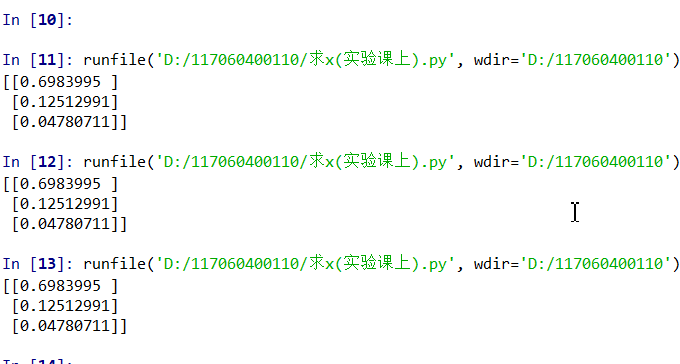
b = np.array([[1,2,3]])

A =np.array([[1,0.5,5],[2.3,2,3],[4,1,1.7]])

x = np.matmul(inv(A),np.transpose(b))

print(x)

**实验结果：**



1. **绘制一副自己的人物肖像的手绘效果图**

**核心代码：**

from PIL import Image

import numpy as np

vec\_el = np.pi/2.2

vec\_az = np.pi/4.

depth = 10.

im = Image.open('4.jpg').convert('L')

a = np.asarray(im).astype('float')

grad = np.gradient(a)

grad\_x,grad\_y = grad

grad\_x = grad\_x\*depth/100.

grad\_y = grad\_y\*depth/100.

dx = np.cos(vec\_el)\*np.cos(vec\_az)

dy = np.cos(vec\_el)\*np.sin(vec\_az)

dz = np.sin(vec\_el)

A = np.sqrt(grad\_x\*\*2 + grad\_y\*\*2 + 1.)

uni\_x = grad\_x/A

uni\_y = grad\_y/A

uni\_z = 1./A

a2 = 255\*(dx\*uni\_x + dy\*uni\_y +uni\_z)

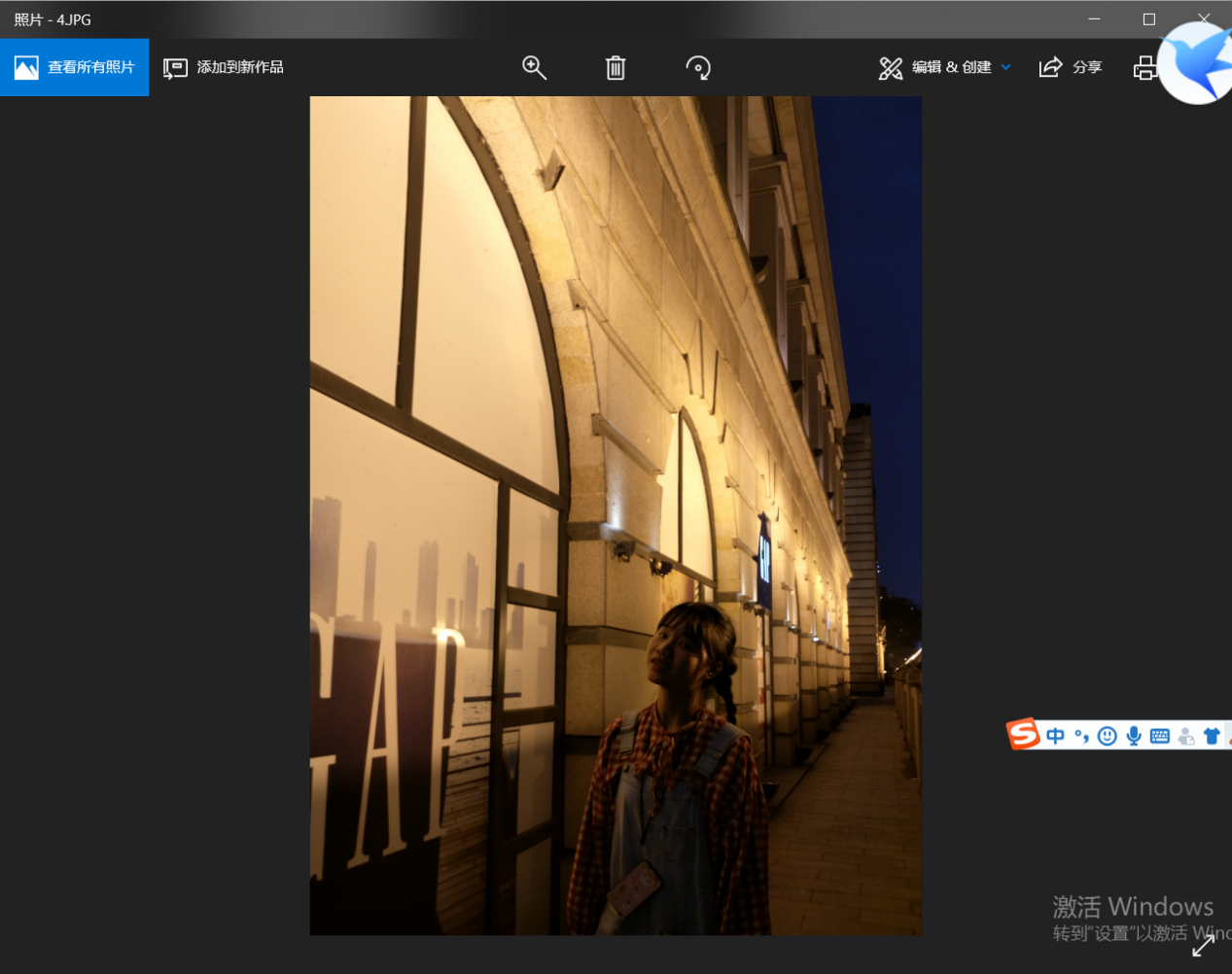
a2 = a2.clip(0,255)

im2 = Image.fromarray(a2.astype('uint8'))

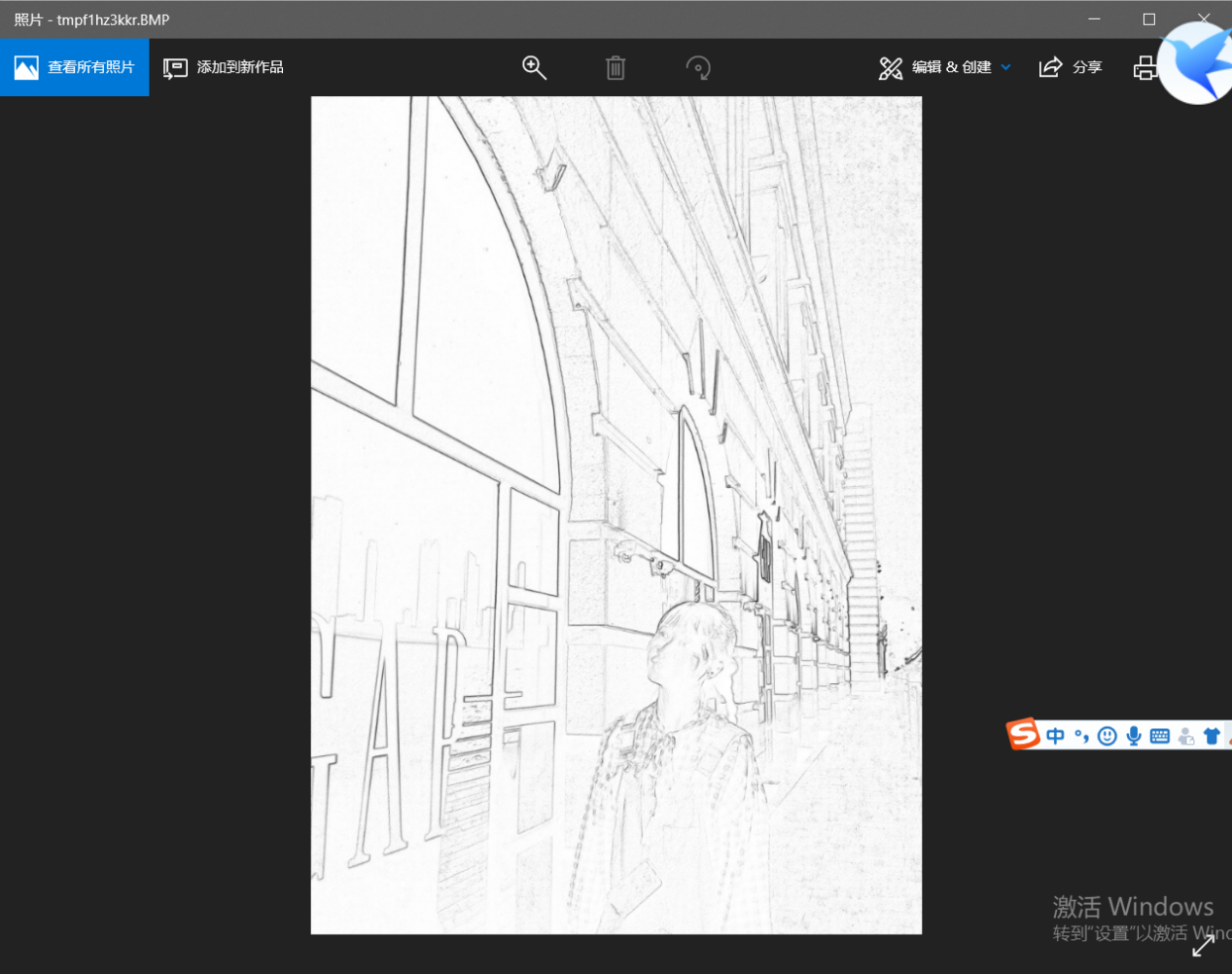
im2.show()

im2.save('fcityHandDraw.jpg')

**原图：**



**实验结果：**



1. **程序练习9-1**

**方波绘制。**

**核心代码：**

（1）

import matplotlib.pyplot as plt

import matplotlib

import numpy as np

t = np.linspace(0,2\*np.pi,100)

k = 1

N = 10

y =0

while k <= 25:

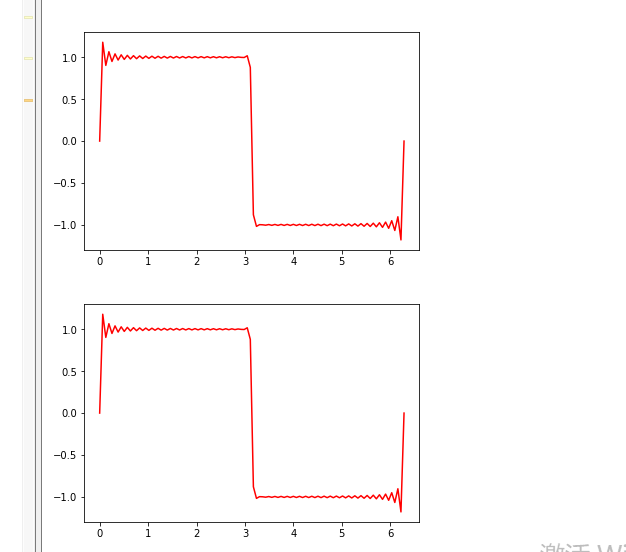
y = y + (4\*np.sin((2\*k-1)\*t))/((2\*k-1)\*np.pi)

k = k + 1

plt.plot(t,y,'r')

plt.show()

**实验结果：**



（2）

import matplotlib.pyplot as plt

import numpy as np

def squareWave(x,n):

f = np.zeros((x.shape[0],))

k = 1

while k <= n:

f = f + (4\*np.sin((2\*k-1)\*x))/((2\*k-1)\*np.pi)

k = k + 1

return f

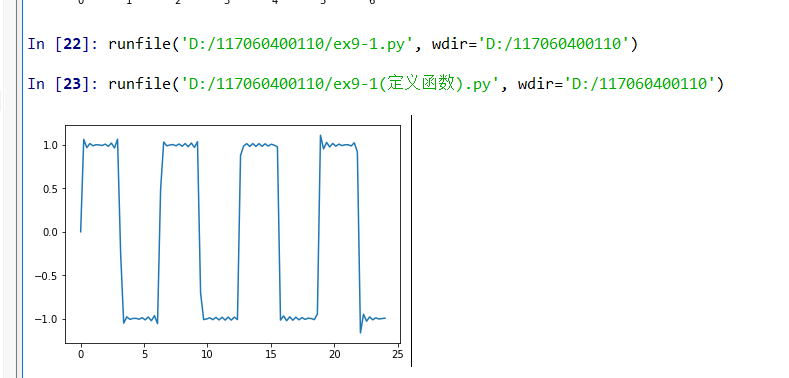
x = np.linspace(0.0,24.0,100)

y = squareWave(x,20)

plt.plot(x,y)

plt.show()

**实验结果：**



1. **圆的绘制**

**（1）**

**核心代码**：

import matplotlib.pyplot as plt

import numpy as np

#t = np.linspace(0,2\*np.pi,100)

x = np.linspace(-1,1,1000)

y = np.linspace(-1,1,1000)

px = []

py = []

for xx in x:

for yy in y:

if np.abs(xx\*\*2 + yy\*\*2 -1) <= 1e-3:

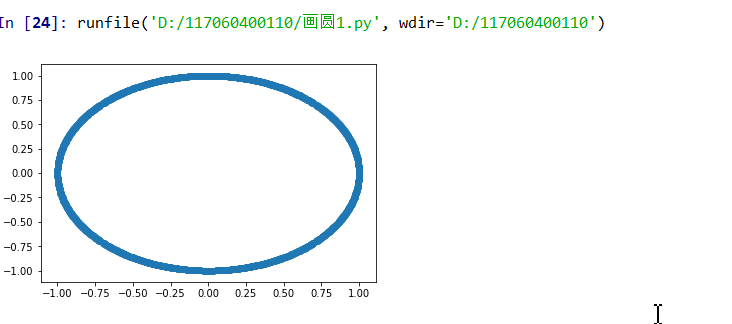
px.append(xx)

py.append(yy)

plt.scatter(px,py)

plt.show()

**实验结果**：



（2）

**核心代码：**

import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(-1,1,1000)

y = np.linspace(-1,1,1000)

px = np.sqrt(1 - x\*\*2)

py = -px

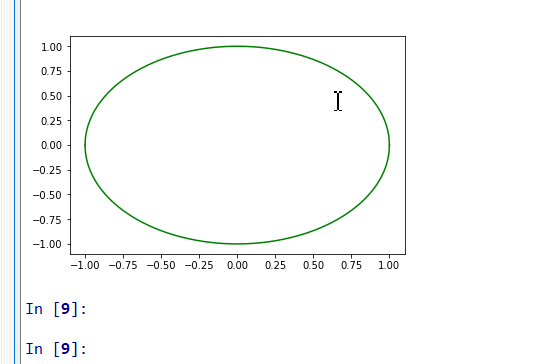
plt.plot(x,px,'g')

plt.plot(x,py,'g')

plot.axis('equal')

plt.show()

实验结果：



1. **matplotlib官网资料的学习**

**（1）源代码原图**

import numpy as np

import matplotlib.pyplot as plt

# evenly sampled time at 200ms intervals

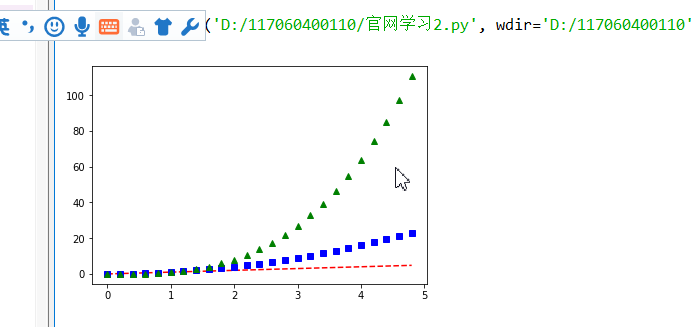
t = np.arange(0., 5., 0.2,)#用于创建等差数组（开始值，终值，步长）

# red dashes, blue squares and green triangles

plt.plot(t, t, 'r--', t, t\*\*2, 'bs', t, t\*\*3, 'g^')#(参数为t,函数，对应函数用什么来绘制出）

plt.show()

图：



**学习：**

**核心代码：**

import numpy as np

import matplotlib.pyplot as plt

# evenly sampled time at 200ms intervals

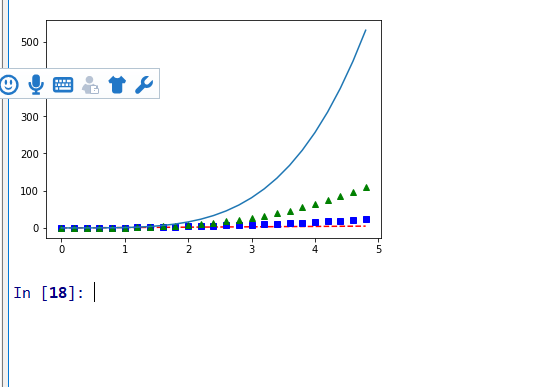
t = np.arange(0., 5., 0.2,)#用于创建等差数组（开始值，终值，步长）

# red dashes, blue squares and green triangles

plt.plot(t, t, 'r--', t, t\*\*2, 'bs', t, t\*\*3, 'g^',t,t\*\*4)(参数为t,函数，对应函数用什么来绘制出）

plt.show()

**结果：**



**实验总结：**

1. 通过此次实验让我更加明白了，图像的矩阵表示和处理
2. 会运用了数据绘制图库进行坐标系绘制
3. 也更加熟悉了函数的定义，如何将代码模块化