实验8 插值、拟合计算与图形用户界面

实验报告

# 1. 实验目的

（1）熟悉使用Python的scipy库插值和拟合运算的方法。

（2）熟悉用 Python语言Tkinter库实现用户界面。

# 2. 实验步骤（详述每一个子实验的关键语法及其实现）

（代码文件与报告同时提交）

2.1 实验任务8-1

主要语法：

1. 在界面上有多个功能区域时，学会使用grid语句实现对界面上控件的布局。
2. 学会使用Label，Entry和Button控件。使用Label设置每行的标签，使用Entry中get方法获取文本与delete方法删除文本。
3. 学会使用窗口变量名**.cget('bg')**的方法完成需要获取主窗口背景色。以及学会改变背景颜色。

代码：

from tkinter import \*

def helloButton():

n1=e1.get()

n2=e2.get()

n3=e3.get()

n4=e4.get()

t1=len(n1)

t2=len(n2)

t3=len(n3)

t4=len(n4)

if n1!=''and n2!=''and n3!=''and n4!='':

e1.delete(0,t1)

e2.delete(0,t2)

e3.delete(0,t3)

e4.delete(0,t4)

print('新注册了一个用户：')

print('用户名是：',n1)

print('密码是:',n2)

print('电子邮箱地址是：',n3)

print('手机号是：',n4)

l5['text']=''

l5['bg']='SystemButtonFace'

else:

l5['text']='有项目未填写！'

l5['bg']='red'

top = Tk()

top.title('注册界面')

height=369

l1=Label(top,text='用户名').grid(row=0)

e1=Entry(top)

e1.grid(row=0,column=1)

l2=Label(top,text='密码').grid(row=1)

e2=Entry(top,show='\*')

e2.grid(row=1,column=1)

l3=Label(top,text='电子邮箱').grid(row=2)

e3=Entry(top)

e3.grid(row=2,column=1)

l4=Label(top,text='手机号').grid(row=3)

e4=Entry(top)

e4.grid(row=3,column=1)

l5=Label(top,text='')

l5.place(x=60,y=95)

l6=Label().grid()

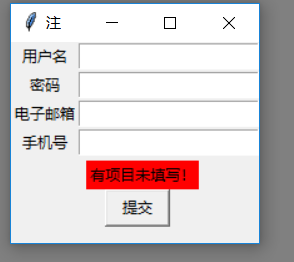
l7=Label().grid()

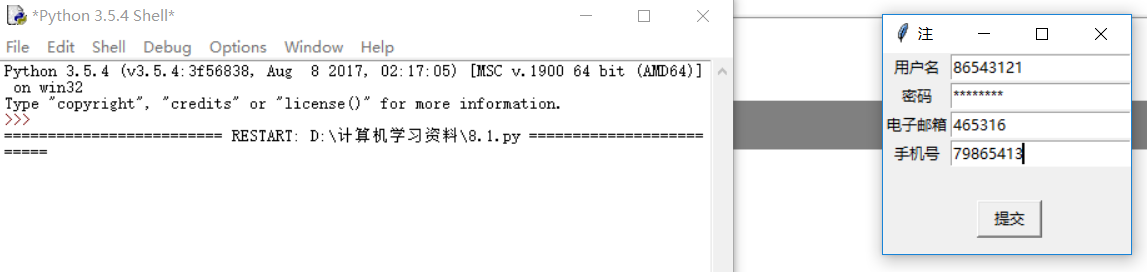
l8=Label().grid()

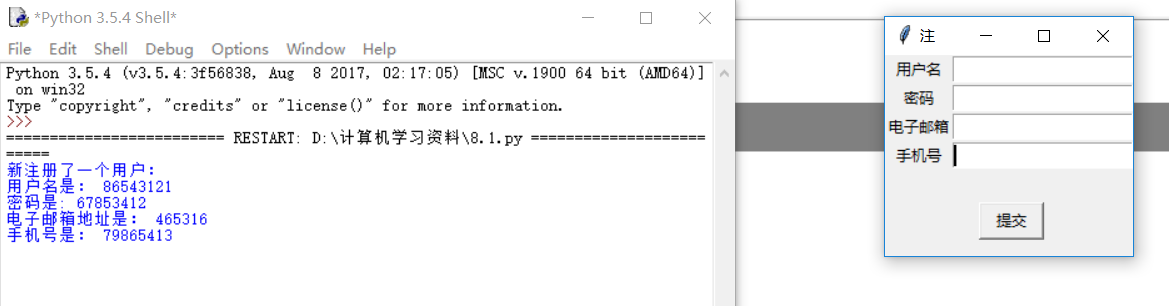
Button(top,text = '提交',anchor='c',width=6,height=1,command = helloButton).place(x=75,y=118)

top.mainloop()

运行结果：







2.2 实验任务8-2

主要语法：

1. 学会创建一个GUI主要包括显示提示信息的若干个 Label，输入参数的若干个 Entry、展示泊松分布与正态分布样本结果的 2 个 Text，以及进行泊松分布数据生成和频率统计的一个 Button
2. 编写定义Button的回调函数，包括
   1. 定义生成符合泊松分布的数的函数；
   2. 随机生成泊松分布样本，并将生成的样本数据插入到一个 Text 中；
   3. 生成正态分布样本，并将生成的样本数据插入到另一个 Text 中；
   4. 统计泊松分布生成的数据生成的频率，进行三次样条插值计算。
3. 学会使用spline函数进行三次样条插值计算。

spline 函数在 scipy.interpolate 包中，格式为yy = spline(x,y,xx) 。 其中 x=[x1,x2,….,xn], y=[y1,y2,…,yn]为样本点（离散的测量数据）。 该函数用三次样条插值计算出由各点(xi, yi)确定的一元函数 y=f(x)在点 xx 处的值 yy。在这个函数中，要求样本点的 x 是从小到大顺序排列的，即 x1<x2<···<xn。

代码：

from tkinter import \*

import math

import random

from scipy import stats

from scipy.interpolate import spline

root = Tk()

root.title('统计')

root.geometry('1200x300')

x11=StringVar()

x12=StringVar()

x21=StringVar()

x22=StringVar()

x23=StringVar()

x11\_entry=Entry(root,width=20,textvariable=x11)

x11\_entry.place(x=260,y=20,anchor=CENTER)

x12\_entry=Entry(root,width=20,textvariable=x12)

x12\_entry.place(x=1100,y=20,anchor=CENTER)

x21\_entry=Entry(root,width=20,textvariable=x21)

x21\_entry.place(x=260,y=40,anchor=CENTER)

x22\_entry=Entry(root,width=20,textvariable=x22)

x22\_entry.place(x=680,y=40,anchor=CENTER)

x23\_entry=Entry(root,width=20,textvariable=x23)

x23\_entry.place(x=1100,y=40,anchor=CENTER)

Label(text='泊松分布样本大小').place(x=80,y=20,anchor=CENTER)

Label(text='泊松分布λ').place(x=920,y=20,anchor=CENTER)

Label(text='正态分布样本大小').place(x=80,y=40,anchor=CENTER)

Label(text='正态分布μ').place(x=500,y=40,anchor=CENTER)

Label(text='正态分布σ').place(x=920,y=40,anchor=CENTER)

Button(text='生成并统计',command=lambda:sheng()).place(x=574,y=66,anchor=CENTER)

Label(text='泊松分布样本').place(x=230,y=92,anchor=CENTER)

Label(text='正态分布样本').place(x=900,y=92,anchor=CENTER)

textbox1= Text(root,width=100,height=13.4)

textbox2= Text(root,width=100,height=13.4)

textbox1.place(x=350,y=192,anchor=CENTER)

textbox2.place(x=950,y=192,anchor=CENTER)

Label(text='λ-0.5插值为：').place(x=70,y=290,anchor=CENTER)

Label(text='λ+1.5插值为：').place(x=400,y=290,anchor=CENTER)

def poisson(a):

k=0

p=math.exp(-a)

s=p

u=random.random()

if u<=math.exp(-a):

return 0

else:

while u>s:

p=a\*p/(k+1)

s=s+p

k=k+1

return k-1

def pin(a,d):

if d in a:

a[d]+=1

else:

a[d]=1

def sheng():

bo1=int(x11.get())

bo2=int(x12.get())

zheng1=int(x21.get())

zheng2=int(x22.get())

zheng3=int(x23.get())

zheng=stats.norm(loc=zheng2,scale=zheng3)

chu=zheng.rvs(size=zheng1)

for i in range (zheng1):

textbox2.insert(INSERT,'%-10d'%(int(chu[i])))

if i%5==4:

textbox2.insert(INSERT,'\n')

tong={}

for i in range (bo1):

flag=poisson(bo2)

textbox1.insert(INSERT,'%-10d'%(flag))

if i%5==4:

textbox1.insert(INSERT,'\n')

pin(tong,flag)

key=list(tong.keys())

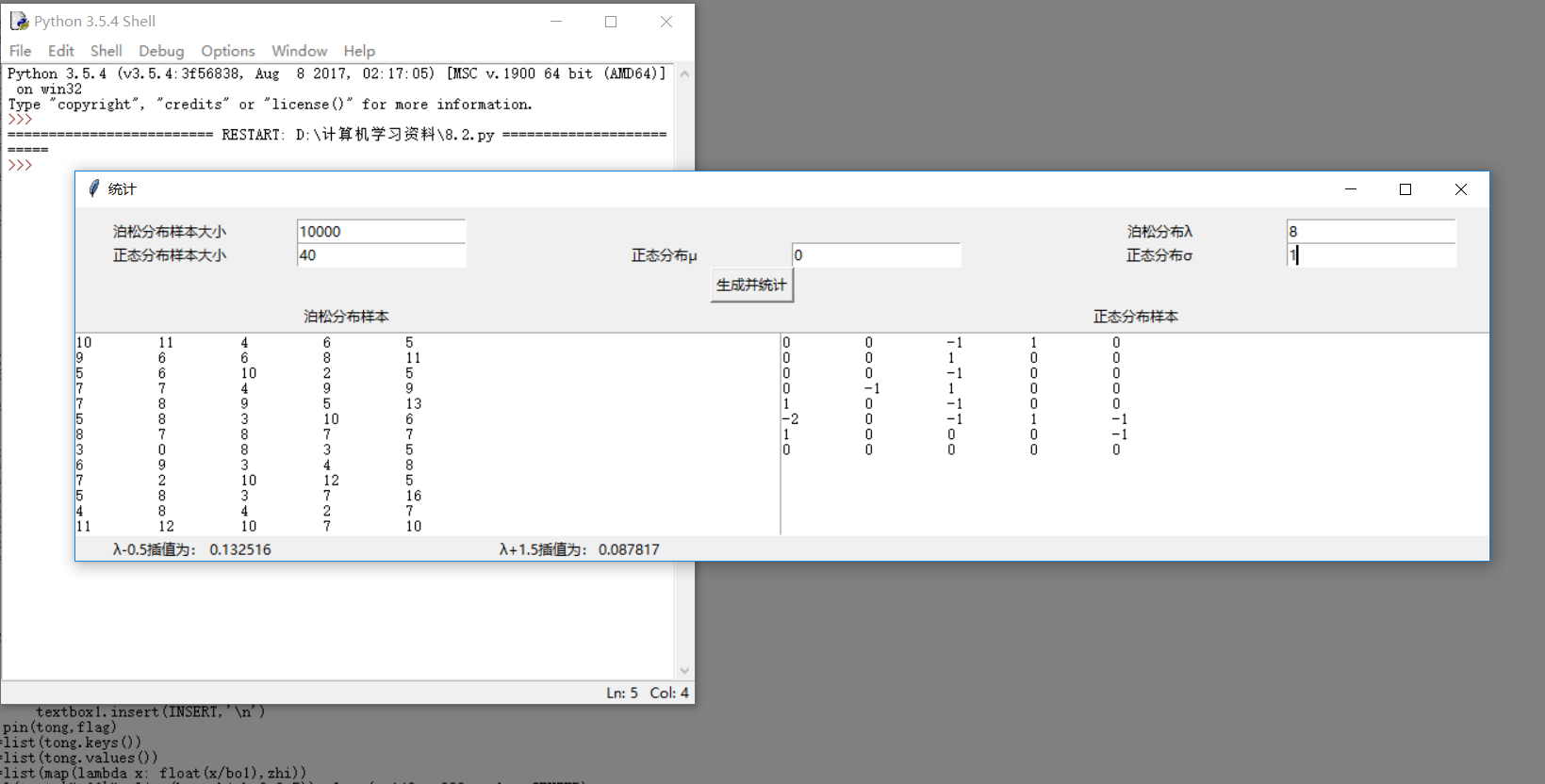
zhi=list(tong.values())

zhi=list(map(lambda x: float(x/bo1),zhi))

Label(text='%.6f'%spline(key,zhi,bo2-0.5)).place(x=140,y=290,anchor=CENTER)

Label(text='%.6f'%spline(key,zhi,bo2+1.5)).place(x=470,y=290,anchor=CENTER)

运行结果：



# 3. 实验总结（实验心得及体会，以及对本课程的建议等）

本次实验难度较高，要求我们自行创建GUI，需要我们读懂课本上的内容，进行模仿创造才能做出，特别的新增了许多语句以及要求我们调用一些Python中的内置安装包，使得我感到非常的陌生，在做起题目来十分吃力。特别的在第二题中需要我们定义生成符合泊松分布的数的函数等函数使得题目难度大大提高，需要大量查阅资料才能将题目做出。在此希望考试时能适当降低题目的难度，使之符合广大同学们的水平，人人取得好成绩。

# 4. 简答题

（1）插值和拟合运算有什么意义？

插值和拟合都是函数逼近或者数值逼近的重要组成部分 ，使得我们在已知的数据中发现规律，实现对未知数据的预测。

（2）如何使用Tkinter建立GUI界面？如何在界面上布局各控件？各控件有什么属性可以设置？请就你的了解作出回答。

创建一个GUI界面：

* 1、导入 Tkinter 模块
* 2、创建控件
* 3、指定这个控件的 master， 即这个控件属于哪一个
* 4、告诉 GM(geometry manager) 有一个控件产生了。

界面上控件的布局可以使用 grid 方法实现，也可使用 pack 方法结合 Frame 控件实现。

控件的属性：

1.Button 按钮。类似标签,但提供额外的功能,例如鼠标掠过、按下、释放以及键盘操作

2.Entry 文本框。单行文字域,用来收集键盘输入(类似 HTML 中的 text)

3.Label 标签。用来显示文字或图片

4.Text 文本域。 多行文字区域,可用来收集(或显示)用户输入的文字。