实验7 科学计算与绘图

实验报告

# 1. 实验目的

（1）学会安装Numpy、Matplotlib等Python扩展模块

（2）利用Numpy和Matplotlib进行科学计算和绘制基本图像

# 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7-1

主要语法:

1.学会使用plt.plot语句， plt.plot(a,b) 其中 a,b 是两个列表， plt.plot 工作的原理是把 a,b 两个列表中对应点画在直角坐标系中，再用直线连接。

2.学会使用arange函数， 横坐标的生成我们可以用 numpy 包中的 arange 函数。 arange(a,b,c) 中 a 为起始点， b 为终止点， c 为步长。返回一个列表（实际类型是 numpy.ndarray）。（也可以使用 linspace 函数）

3.学会使用np.语句绘制函数图像， np.sin 函数会作用于一个列表，对该列表中每个元素进行 sin 运算。（math.sin函数只能作用于一个数）。

4学会使用 plt 一些其他函数设置 title， label,对图像大小限制等。

代码：

import matplotlib.pyplot as plt

import numpy as np

plt.figure(figsize=(8,6))

x=np.linspace(1.0,10.0)

y1=np.sin(1/x)

y2=np.log(x)

z1=plt.plot(x,y1,'r-',linewidth=3,label='sin(1/x)')

z2=plt.plot(x,y2,'g--',linewidth=3,label='log(x)')

plt.xlim(1,10)

plt.ylim(-1.2,4.5)

plt.xlabel('Time(s)')

plt.title('volt')

plt.ylabel('CHT figure')

plt.legend(['sin(1/x)','log(x)'])

plt.show()

运行结果：

图片包含 屏幕截图

已生成高可信度的说明

2.2 实验任务7-2

主要语法：1.学会编写产生符合泊松分布随机数的代码

2.学会导入math和random模块，解决问题

3. 学会导入spicy模块，使用其中的特殊函数。正态分布部分需要使用 SciPy 模块，所需要的各个数学特征都可通过scipy.randn()以及 scipy.stats.describe()完成。

代码：

import math

import random

from scipy import stats

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

print('Poisson distribution:')

for i in range(8):

x=poisson(8)

print(x,end=' ')

print('')

print('')

print('Normal distrubution:')

print('Number of elements:10')

print('Elements:')

a=stats.norm(loc=0,scale=1)

b=a.rvs(size=10)

print(b)

print('Minimum:%.5f Maximum:%.5f'%(stats.describe(b).minmax[0],stats.describe(b).minmax[1]))

print('Mean:%.5f'%stats.describe(b).mean)

print('Variance:%.5f'%stats.describe(b).variance)

print('Skewness:%.5f'%stats.describe(b).skewness)

print('Kurtosis:%.5f'%stats.describe(b).kurtosis)

运行结果：

Poisson distribution:

14 4 2 8 9 8 8 4

Normal distrubution:

Number of elements:10

Elements:

[ 1.12254277 2.43539719 0.4865446 -0.35779125 1.51309538 -0.15293614

-0.08686946 2.27426098 1.39488535 -1.48117697]

Minimum:-1.48118 Maximum:2.43540

Mean:0.71480

Variance:1.56433

Skewness:-0.19315

Kurtosis:-0.93725

2.3 实验任务7-3

主要语法：

1.学会绘制柱状图

2.学会使用bar函数绘制柱状图

代码：

import matplotlib.pyplot as plt

import numpy as np

from pylab import mpl

mpl.rcParams['font.sans-serif'] = ['SimHei']

shuju=[[u'A组数据',96,67,89,65],[u'B组数据',56,37,59,64],[u'C组数据',78,89,67,5],

[u'D组数据',87,47,37,24],[u'E组数据',78,36,34,76],[u'F组数据',56,49,76,87],

[u'G组数据',86,32,23,34],[u'H组数据',65,25,27,65],[u'I组数据',52,97,75,24]]

ind=np.arange(9)

shuju1Mean=(96,56,78,87,78,56,86,65,52)

shuju2Mean=(67,37,89,47,36,49,32,25,97)

shuju3Mean=(89,59,67,37,34,76,23,27,75)

shuju4Mean=(65,64,45,24,76,87,34,65,24)

width=0.23

fig,ax=plt.subplots()

rects1=ax.bar(ind,shuju1Mean,width,color='blue')

rects2=ax.bar(ind+width,shuju2Mean,width,color='yellow')

rects3=ax.bar(ind+2\*width,shuju3Mean,width,color='green')

rects4=ax.bar(ind+3\*width,shuju4Mean,width,color='red')

ax.legend((rects1[0],rects2[0],rects3[0],rects4[0]),(u'附加数据1',u'附加数据2',u'附加数据3',u'附加数据4'))

def autolabel(rects):

for rect in rects:

height=rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2.,1.05\*height,'%d'%int(height),ha='center',va='bottom')

autolabel(rects1)

autolabel(rects2)

autolabel(rects3)

autolabel(rects4)

ax.set\_ylabel(u'数值')

ax.set\_title(u'数据分析表')

ax.set\_xticks(ind+width)

ax.set\_xticklabels((u'A组数据',u'B组数据',u'C组数据',u'D组数据',u'E组数据',u'F组数据',u'G组数据',u'H组数据',u'I组数据'))

plt.show()

运行结果：

图片包含 书写用具, 定居者

已生成极高可信度的说明

# 3. 实验总结（实验心得及体会，以及对本课程的建议等）

这次实验难度相比以往有所降低，只是需要我们看懂课本进行模仿大致就能完成，但第二题有点难度，我看不懂题意，没做出来。总而言之，编程是需要勤加练习的，这样我们才能得心应手的将题目做出。

# 4. 简答题

（1）从柱状图、折线图、饼图、散点图、雷达图中任选两种，简述其使用matplotlib进行绘制的基本步骤。

柱状图：

1. 设置不同的填充模式
2. 调用bar函数
3. 调用pyplot中的legend的方法
4. 定义自动标注函数

折线图：

1.调用plot方法

（2）数据可视化还有什么方式?

一、面积&尺寸可视化  
对同一类图形（例如柱状、圆环和蜘蛛图等）的长度、高度或面积加以区别，来清晰的表达不同指标对应的指标值之间的对比。这种方法会让浏览者对数据及其之间的对比一目了然。制作这类数据可视化图形时，要用数学公式计算，来表达准确的尺度和比例。  
二、颜色可视化  
通过颜色的深浅来表达指标值的强弱和大小，是数据可视化设计的常用方法，用户一眼看上去便可整体的看出哪一部分指标的数据值更突出。  
三、图形可视化  
在我们设计指标及数据时，使用有对应实际含义的图形来结合呈现，会使数据图表更加生动的被展现，更便于用户理解图表要表达的主题。  
四、地域空间可视化  
当指标数据要表达的主题跟地域有关联时，我们一般会选择用地图为大背景。这样用户可以直观的了解整体的数据情况，同时也可以根据地理位置快速的定位到某一地区来查看详细数据。  
五、概念可视化  
通过将抽象的指标数据转换成我们熟悉的容易感知的数据时，用户便更容易理解图形要表达的意义。