**课程实践作业六实验报告**

1. **数据统计程序**

**该程序首先需要读入存在xls文件中的数据，这需要用到扩展包xlrd。出于扩展性，我们不硬性规定数据组数，而是通过读取列数来获知数据数量。然后通过循环运算计算出平均值，方差相关系数等数据，并计算出其线性回归直线。在此基础上，利用扩展包matplotlib绘出散点图和回归直线。具体代码如下：**

import xlrd

import matplotlib.pyplot as plt

import numpy as np

filename = *'testData.xlsx'*

excel = xlrd.open\_workbook(filename)

sheet = excel.sheet\_by\_index(0)

data\_num = sheet.ncols/2

xTotal = 0

yTotal = 0

counter=1

for i in range(0,data\_num\*2,2):

tempx=[]

tempx = sheet.col\_values(i)

j= i+1

tempy = []

tempy = sheet.col\_values(j)

for k in range(len(tempx)):

xTotal += float(tempx[k])

yTotal += float(tempy[k])

tempx[k] = float(tempx[k])

tempy[k] = float(tempy[k])

xTotal = xTotal/len(tempx)

yTotal = yTotal/len(tempy)

varianceX = 0

varianceY = 0

mulXY=0

for k in range(len(tempx)):

varianceX += pow(tempx[k]-xTotal,2)

varianceY += pow(tempy[k]-yTotal,2)

mulXY+= (tempx[k]-xTotal)\*(tempy[k]-yTotal)

r=mulXY/pow(varianceX\*varianceY,0.5)

print(*"The average value of the x&y in the "*+ str(counter)+ *" th data is:"*+str(xTotal)+*" & "*+str(yTotal)+

*", \nand the variance of the x&y corresponding is: "*+str(varianceX)+*" & "*+str(varianceY)+

*"\nand the relevance coefficient is: "*+str(r))

sumXY=0;

sumXX=0;

for k in range(len(tempx)):

sumXY+= tempx[k]\*tempy[k]

sumXX+= tempx[k]\*tempx[k]

a=(xTotal\*len(tempx)\*yTotal-sumXY)/(xTotal\*xTotal\*len(tempx)-sumXX)

b=yTotal-a\*xTotal

plt.figure(counter)

plt.plot(tempx,tempy,*'o'*)

x = np.linspace(1,20,1000)

y = a\*x+b

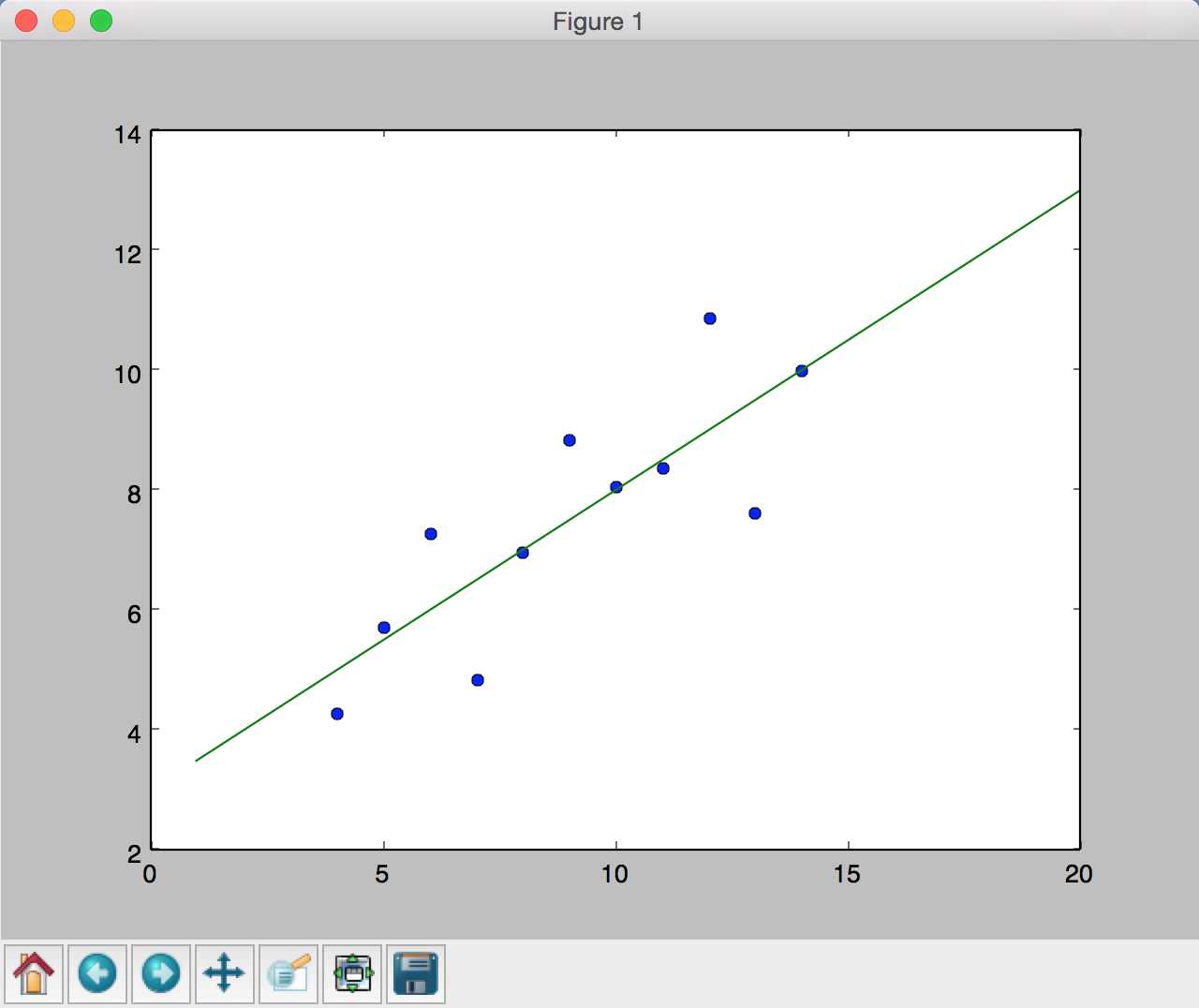
plt.plot(x,y)

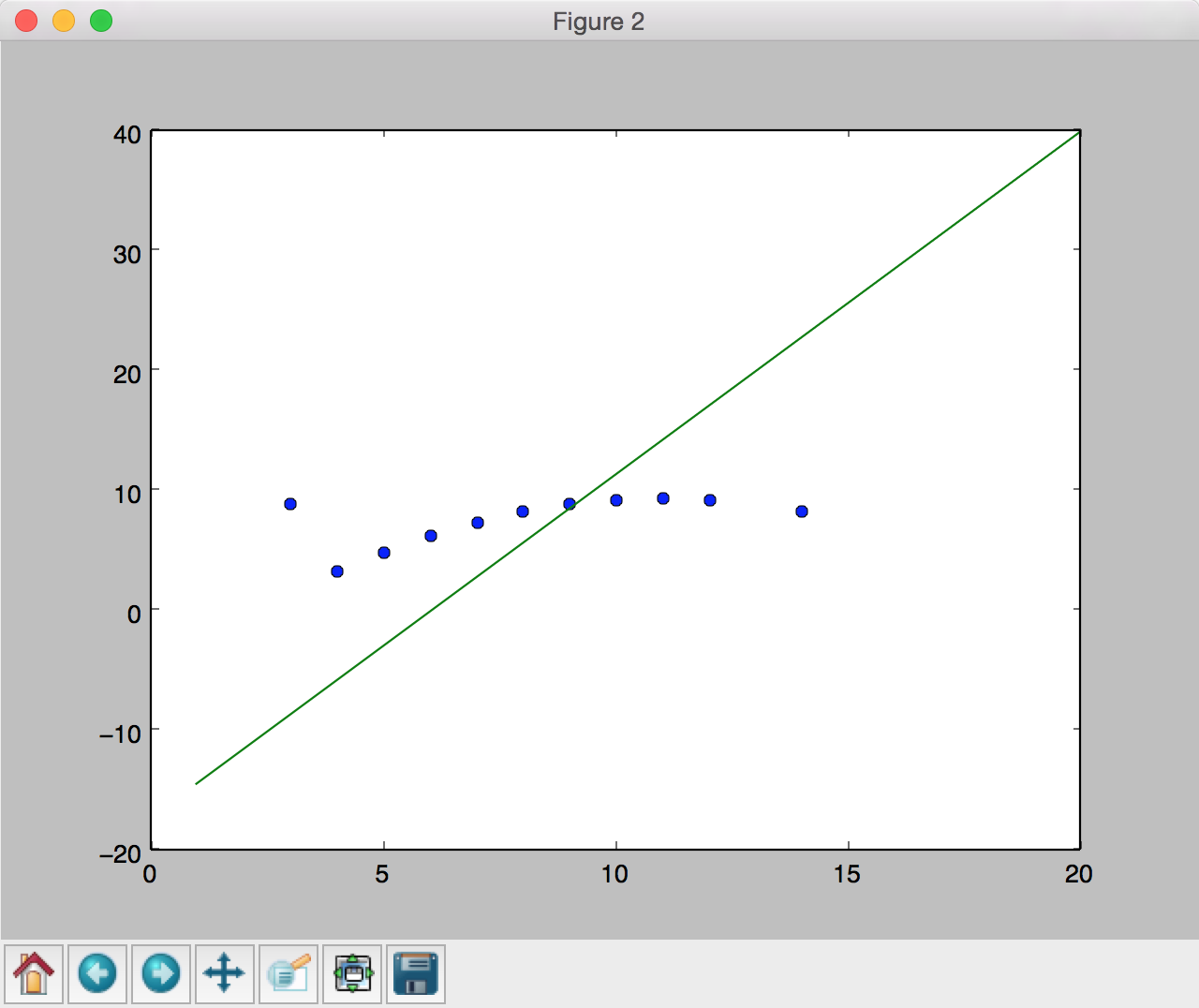
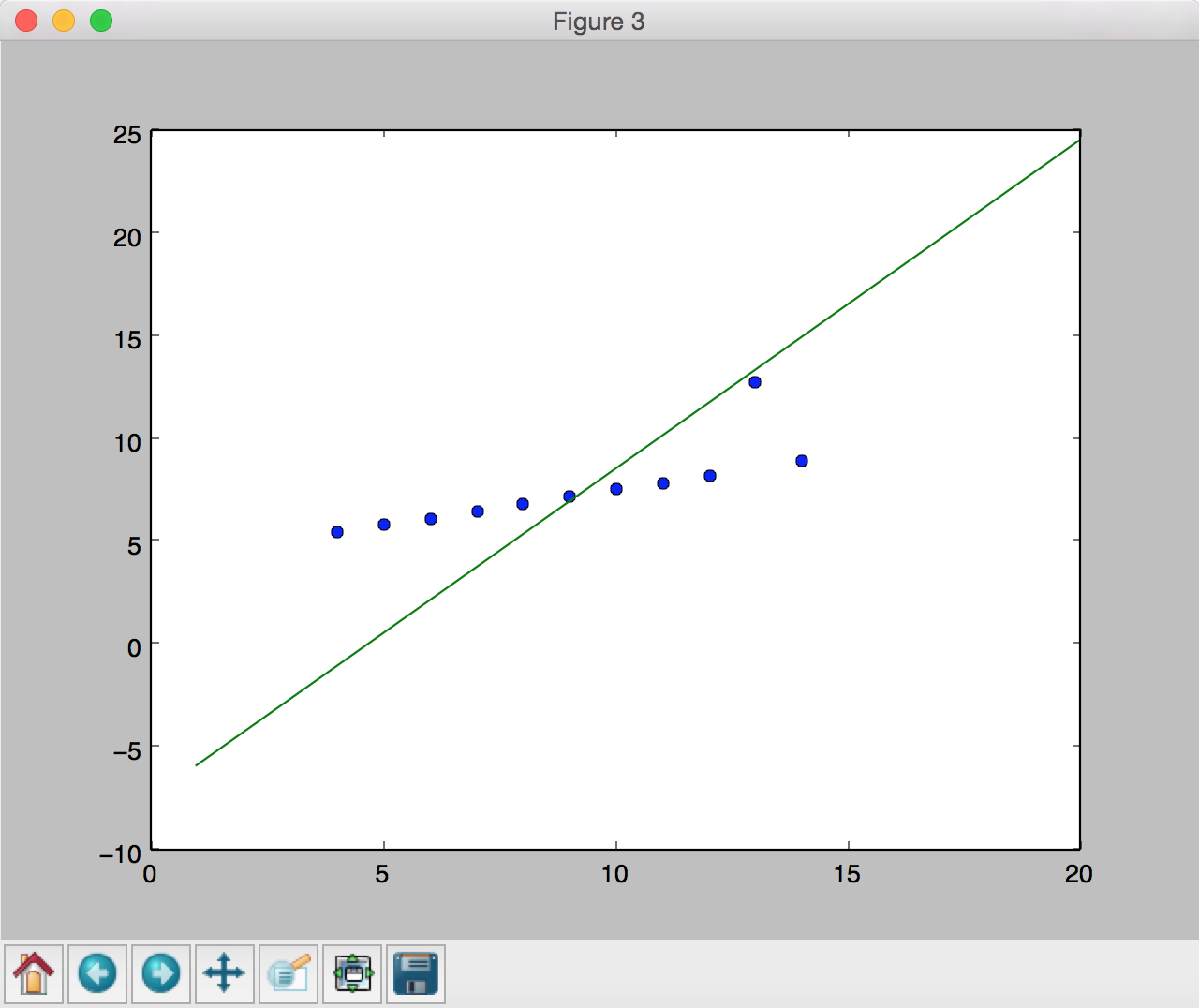
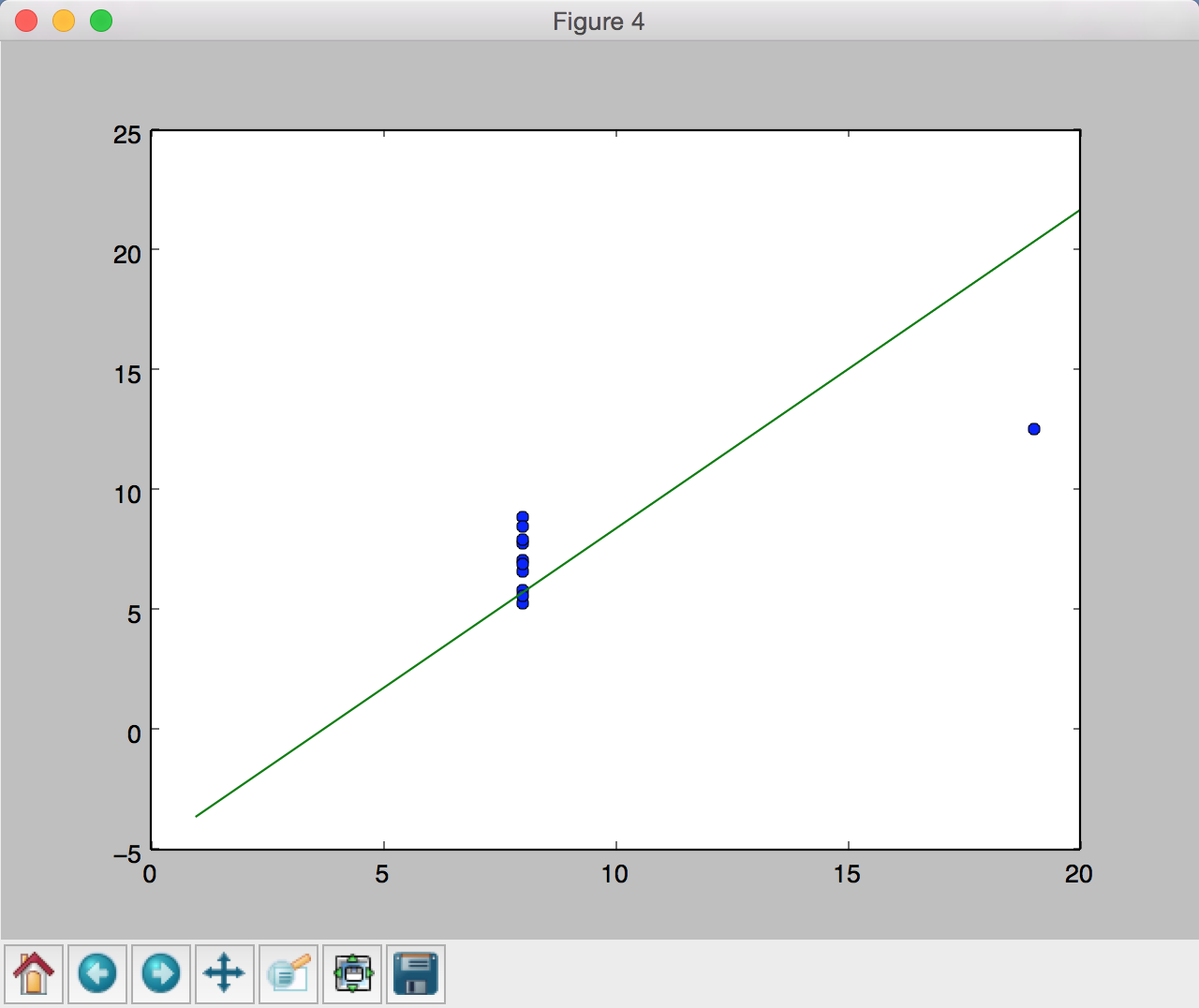
counter+=1

plt.show()

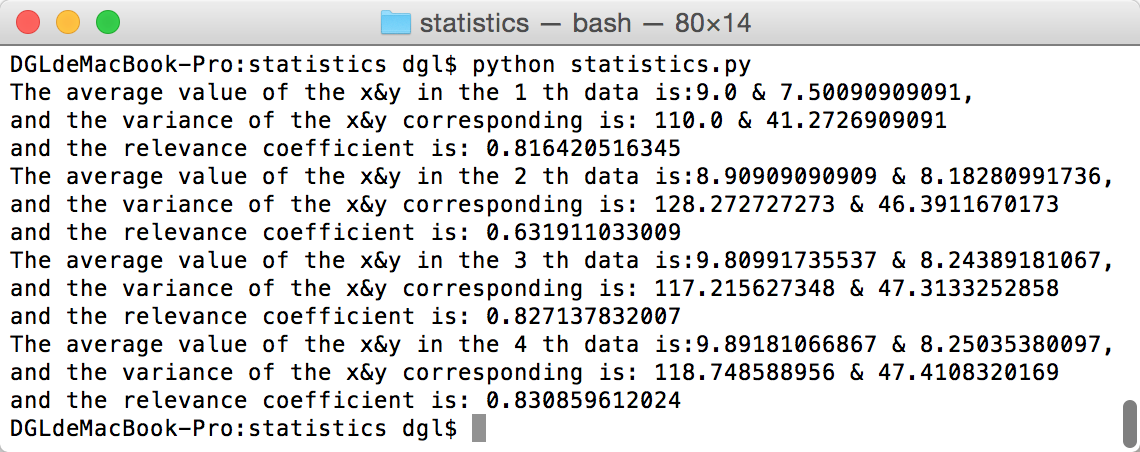
1. **程序结果展示**

**首先展示四幅绘制出来的图片，分别对应四组数据：**

****

**** **** 

**接下来是在控制台输出的关于数据平均值和方差以及相关系数这三个数据：**

****

**首先对于这次试验主要使用的包是numpy和画图的matplotlib，我并没有用到statistics中的计算函数，而是自己计算了方差平均值等等，当然用工具包是更加快捷和准确的方式。总体来说我认为没有太大难度，重点还是放在熟悉各类包上。**

**对于这次试验讲一下我的感受，对比C++这样的语言，python感觉更加像matlab，使用起来会有很多方便的工具包，同时网上说python效率将近C++的70%，这算是比较高的效率了，同时python的可视化比较成熟。作为脚本语言，个人认为开源是其生命所在，因此实验给我最大的收获应该是对数据统计包做了更深入的了解。以上。**