**课程实践作业六**

编写数据统计分析交互计算IPython Notebook

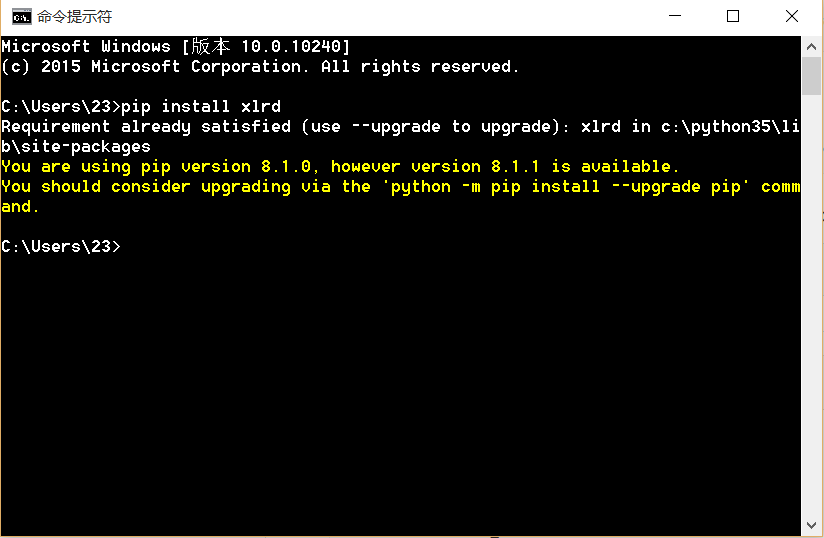
问题：

16.4 Statistical Measures Don't, Tell the Whole Story

In 1973, the statistician F.J. Anscombe published a paper containing the table below. It contains the <x, y> coordinates of the points in each of four data sets.

## 1.插件准备

根据指导书给出的提示可知，本次作业的目的是要对Excel里的数据进行分析，所以需要安装可以从Excel中提取数据的xlrd插件。将数据可视化需要Matplotlib插件。

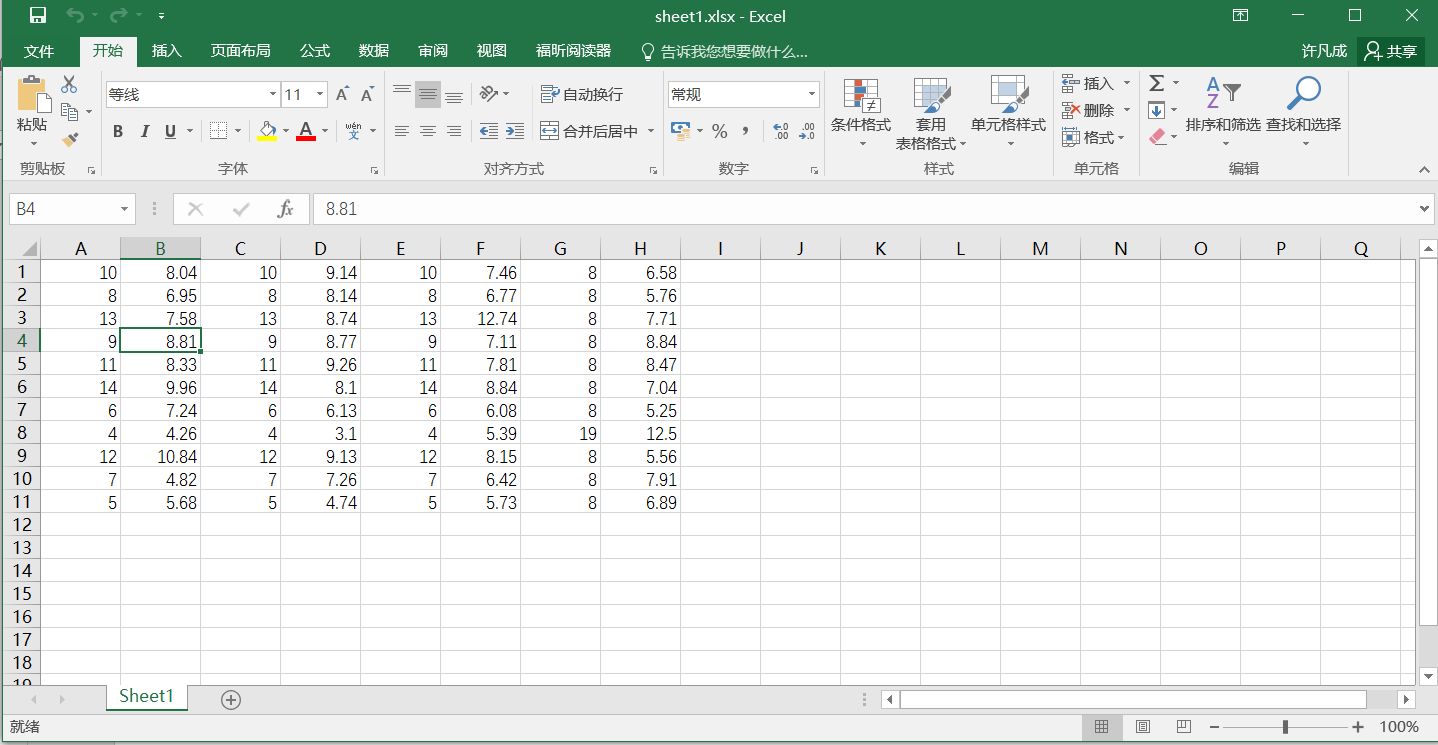


## 2.程序编写

本次作业可以分为三个步骤，首先从Excel中读取数据，然后对四组数据分别进行数学统计，最后完成数学回归并绘制图表。下面对这三个步骤进行详细说明。

### 2.1从excel中读取列表数据

首先我们需要将数据写入Excel中，并为该文件取一个英文名，并且Excel保存的位置也不要出现中文或纯数字，否则python很可能会找不到文件。

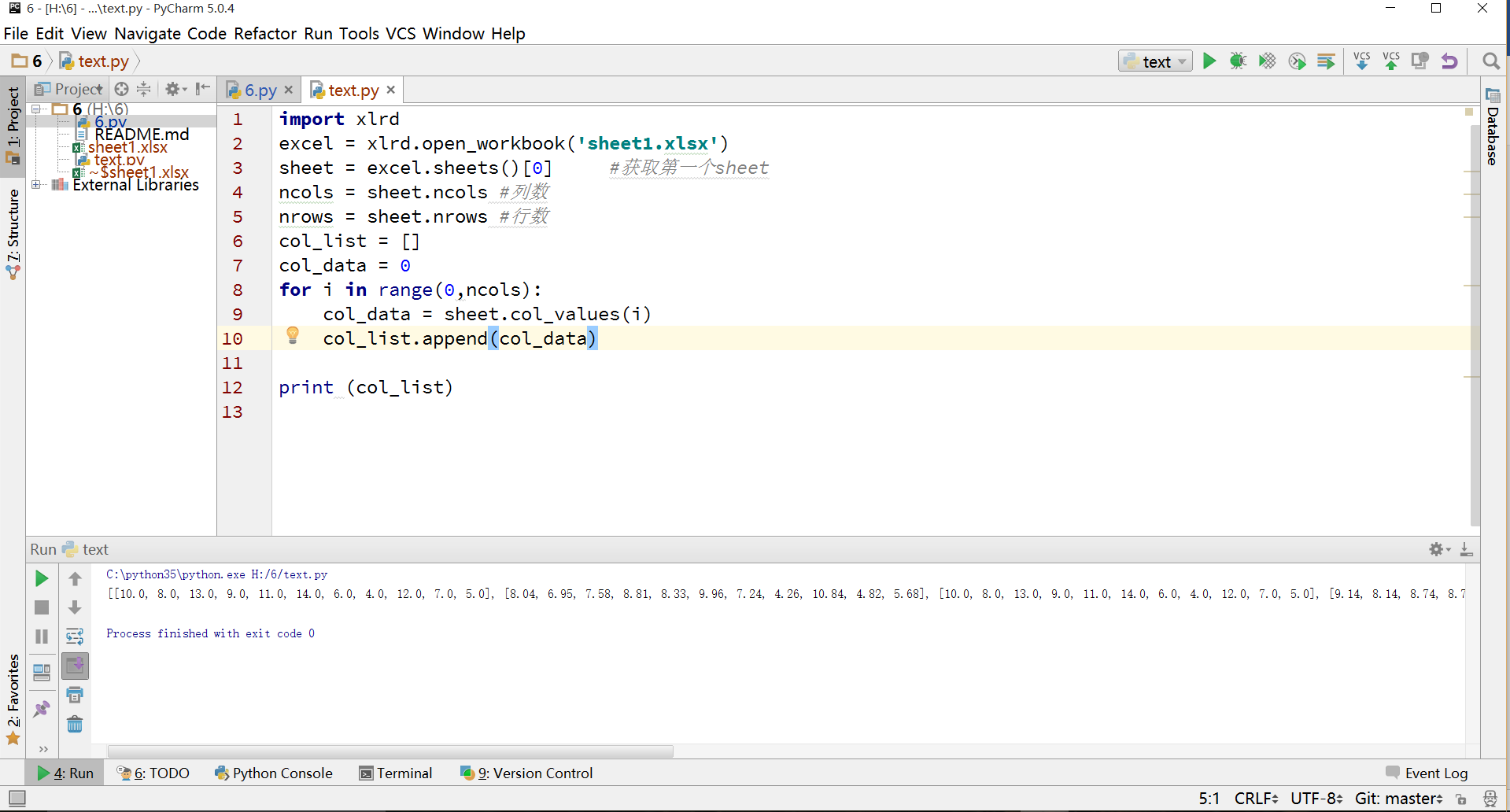


上网查询从Excel中读取数据的方法：

<http://www.jb51.net/article/42635.htm>

通过学习与理解，在pycharm中编写了如下的测试代码：

import xlrd  
excel = xlrd.open\_workbook('sheet1.xlsx')  
sheet = excel.sheets()[0] #获取第一个sheet  
ncols = sheet.ncols #列数  
nrows = sheet.nrows #行数  
col\_list = []  
col\_data = 0  
for i in range(0,ncols):  
 col\_data = sheet.col\_values(i)  
 col\_list.append(col\_data)



运行结果与Excel中的数据相同，表明数据已导入成功。

### 2.2四组数据分别进行数学统计

接下来使用numpy插件对刚才读取的数据进行统计计算，会使用到以下函数：

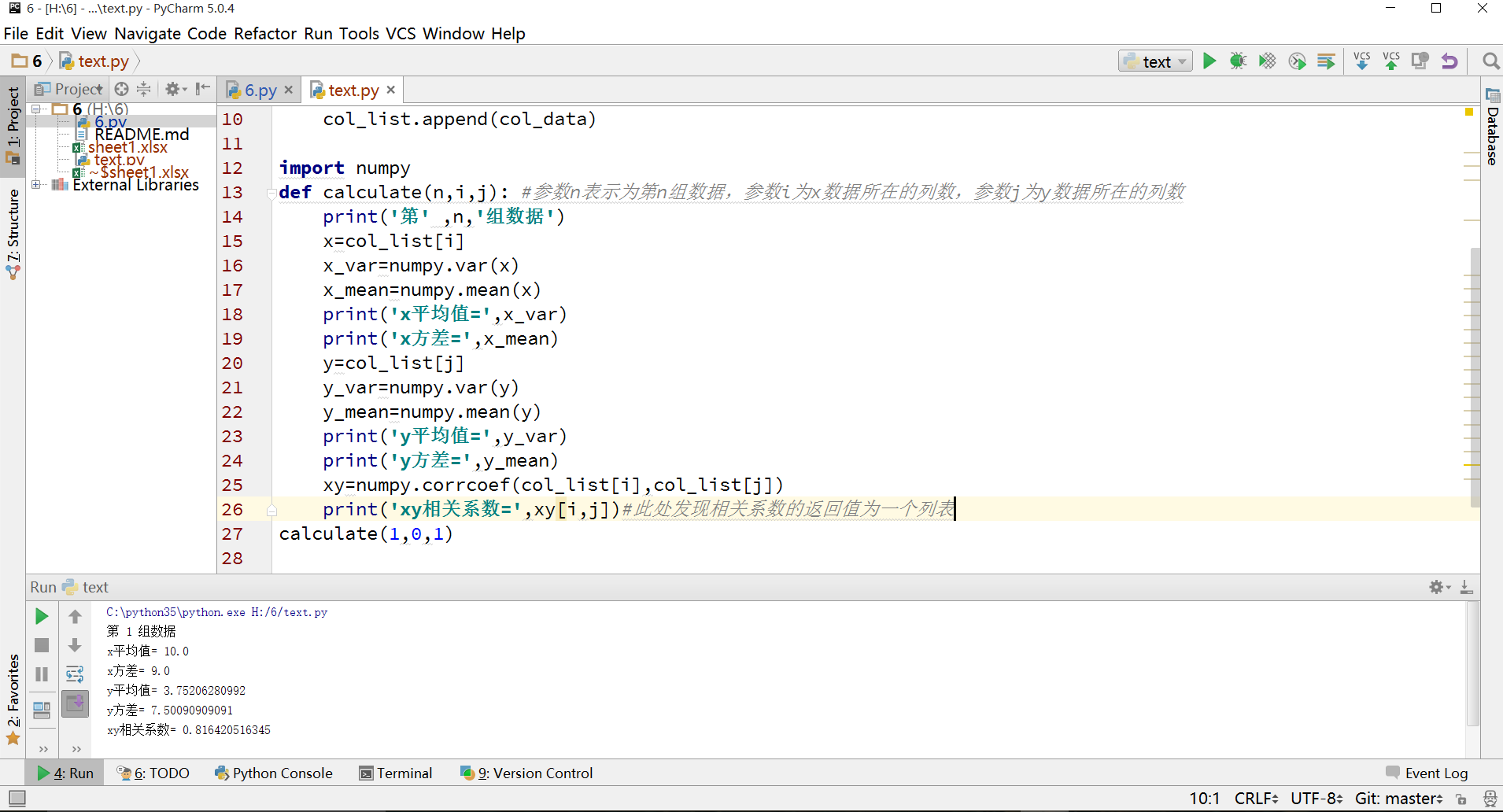
平均值 numpy.var( )

方差 numpy.mean( )

相关系数 numpy.correlate( , )

考虑到每两列数据（一列x，一列y）为一组要分析的数据，故可以先编写一个可以计算平均值，方差，相关系数计算的函数。然后再讲此函数循环四次，即可得到全部数据的计算结果。编写代码如下：

import numpy  
def calculate(n,i,j): #参数n表示为第n组数据，参数i为x数据所在的列数，参数j为y数据所在的列数  
 print('第' ,n,'组数据')  
 x=col\_list[i]  
 x\_var=numpy.var(x)  
 x\_mean=numpy.mean(x)  
 print('x平均值=',x\_var)  
 print('x方差=',x\_mean)  
 y=col\_list[j]  
 y\_var=numpy.var(y)  
 y\_mean=numpy.mean(y)  
 print('y平均值=',y\_var)  
 print('y方差=',y\_mean)  
 xy=numpy.corrcoef(col\_list[i],col\_list[j])  
 print('xy相关系数=',xy[0,1])#此处发现相关系数的返回值为一个列表



然后加入循环，这里要注意列表的下标是从0开始计数。



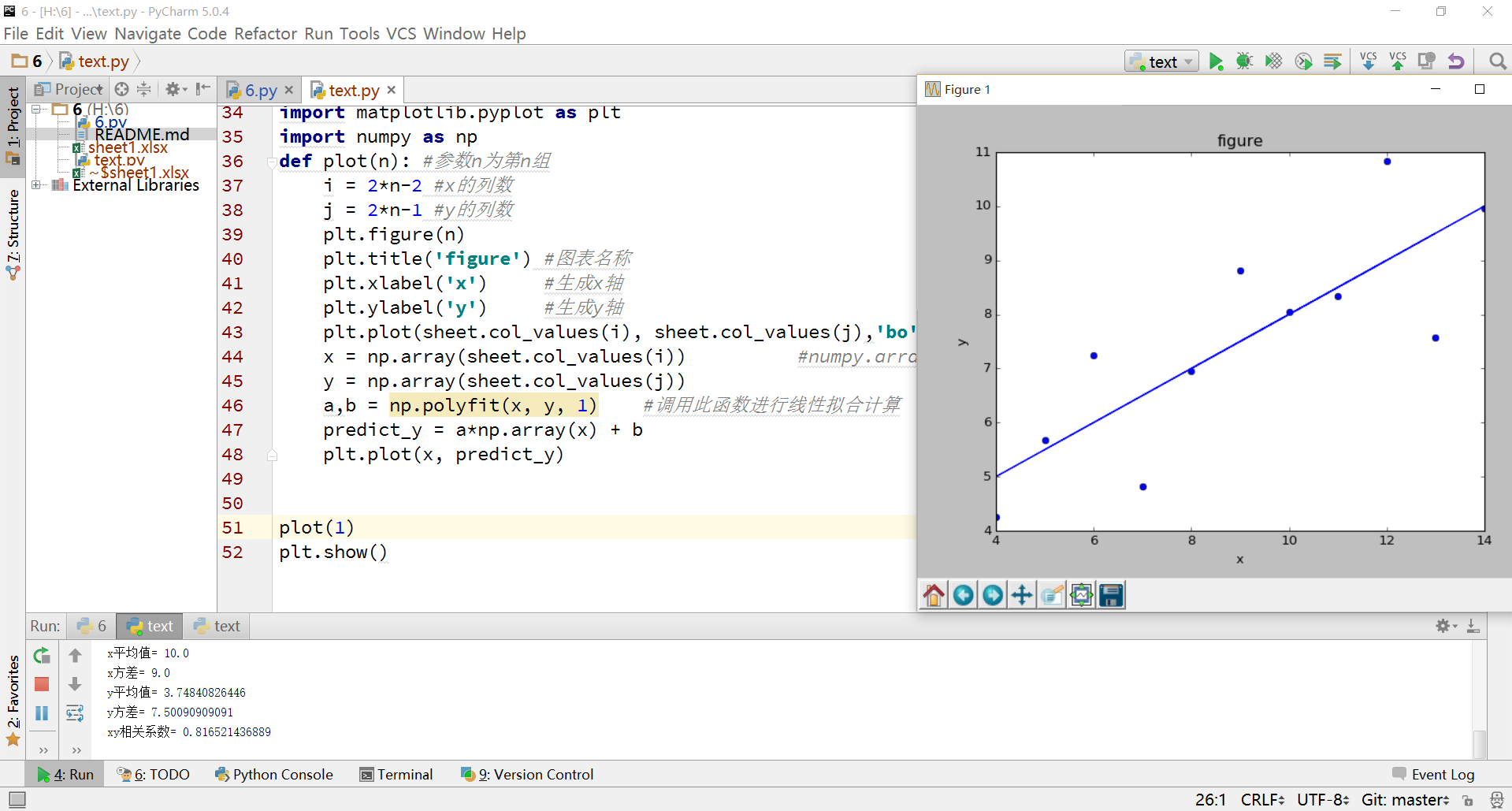
### 2.3完成线性回归与图表绘制

数据计算完毕后，需要对数据进行线性回归并绘制图表，此处采用Matplotlib。相关的操作说明可以参考notebook11 PLOTTING AND MORE ABOUT CLASSES 和Notebook1515 UNDERSTANDING EXPERIMENTAL DATA

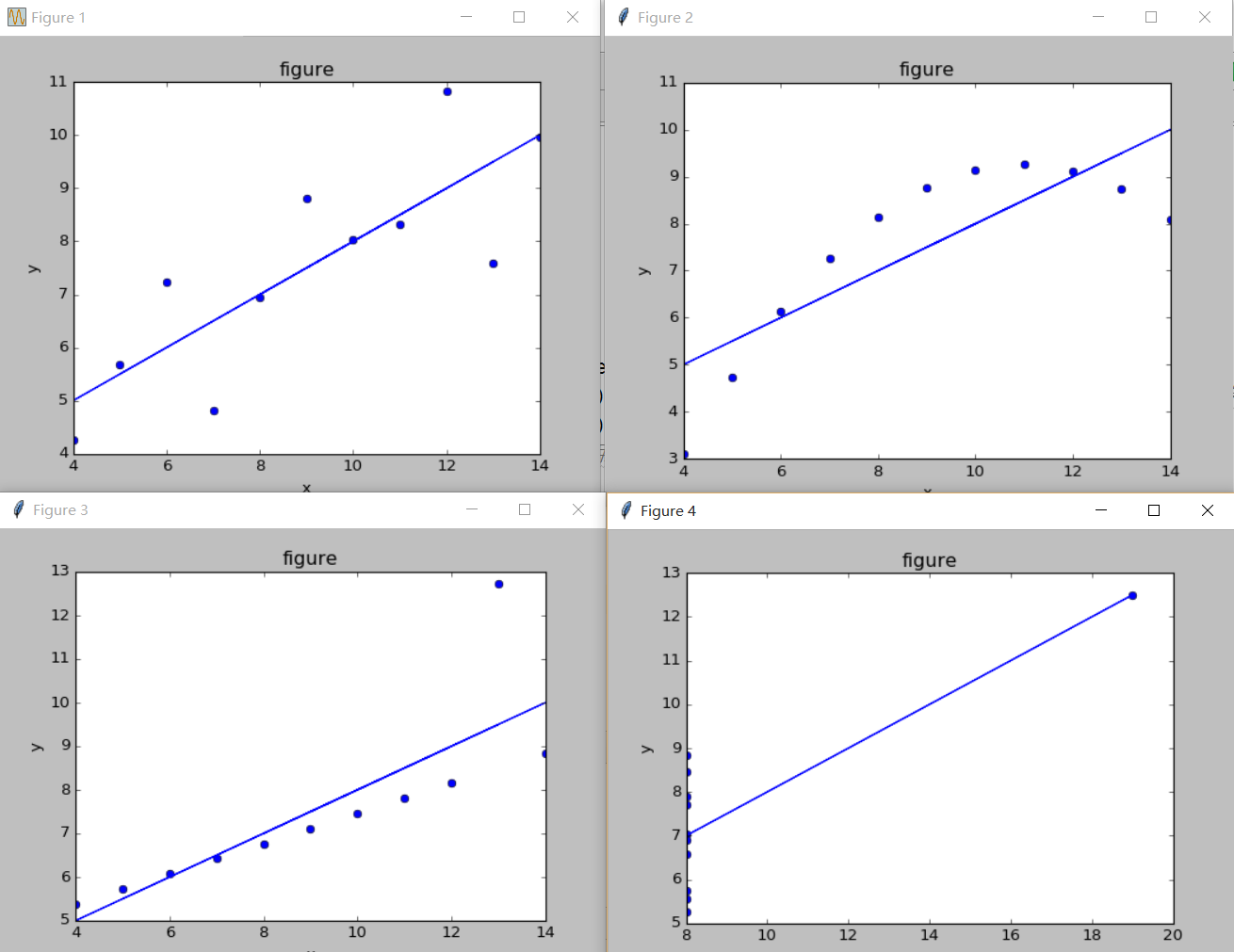
编程思路同2.2，首先对一组数据进行绘图，然后循环四次。

程序及运行结果如下：

import matplotlib.pyplot as plt  
import numpy as np  
def plot(n): #参数n为第n组  
 i = 2\*n-2 #x的列数  
 j = 2\*n-1 #y的列数  
 plt.figure(n)  
 plt.title('figure') #图表名称  
 plt.xlabel('x') #生成x轴  
 plt.ylabel('y') #生成y轴  
 plt.plot(sheet.col\_values(i), sheet.col\_values(j),'bo') #作散点图  
 x = np.array(sheet.col\_values(i)) #numpy.array（）可以进行类型转换，将列表转换为数组类型  
 y = np.array(sheet.col\_values(j))  
 a,b = np.polyfit(x, y, 1) #调用此函数进行线性拟合计算  
 predict\_y = a\*np.array(x) + b  
 plt.plot(x, predict\_y)



加入循环后可得到四幅图：



## 3.小结

通过本次作业，我对如何使用python解决一些实际问题有了些了解。对于初学者经常遇到的一些问题，网络上一般都会有解决方案。学会查找资料并合理运用会对我们的学习有很大的帮助。