# 《Python程序设计综合训练》

# 实验报告

# 班级：

# 姓名：

# 学号：

# 目录

[实验一：爬虫 2](#_Toc137302452)

[1. 实验内容 2](#_Toc137302453)

[2. 实验过程及结果 2](#_Toc137302454)

[3. 实验中遇到的问题及解决 5](#_Toc137302455)

[4. 实验总结 5](#_Toc137302456)

[实验二：办公自动化 6](#_Toc137302457)

[1. 实验内容 6](#_Toc137302458)

[2. 实验过程及结果 6](#_Toc137302459)

[3. 实验中遇到的问题及解决 10](#_Toc137302460)

[4. 实验总结 10](#_Toc137302461)

[实验三：科学计算 10](#_Toc137302462)

[1. 实验内容 10](#_Toc137302463)

[2. 实验过程及结果 10](#_Toc137302464)

[3. 实验中遇到的问题及解决 14](#_Toc137302465)

[4. 实验总结 14](#_Toc137302466)

[实验四：可视化 15](#_Toc137302467)

[1. 实验内容 15](#_Toc137302468)

[2. 实验过程及结果 15](#_Toc137302469)

[3. 实验中遇到的问题及解决 21](#_Toc137302470)

[4. 实验总结 22](#_Toc137302471)

[实验五：自动化选课系统（未完善） 22](#_Toc137302472)

[1. 实验内容 22](#_Toc137302473)

[2. 实验过程及结果 22](#_Toc137302474)

[3. 实验总结 23](#_Toc137302475)

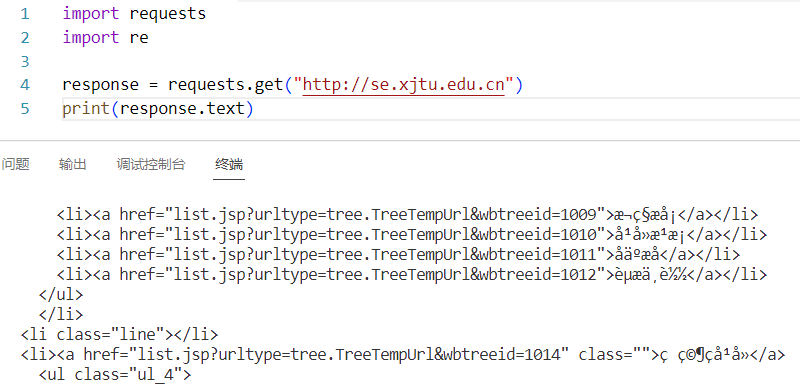
## 实验一：爬虫

### 实验内容

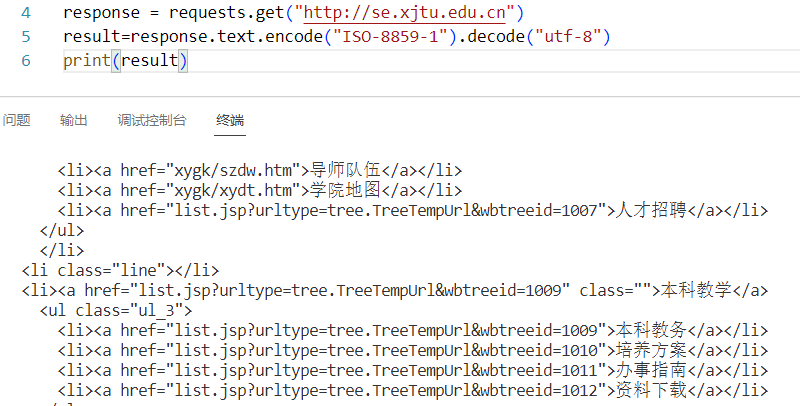
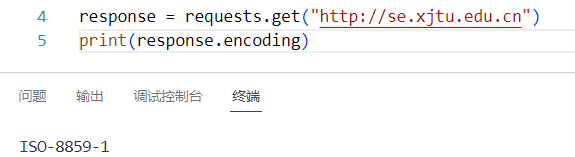
利用requests、selenium模块，实现获取网页源代码、解析网页源代码数据、保存目标数据。

### 实验过程及结果

requests模块的get函数对指定的网址发起请求，服务器根据请求的网址返回一个response对象。response对象的text属性中存储了网页的源代码，但直接将其输出时其中的中文会出现乱码，如图所示。



这是网页源代码的编码与输出编码不同所致。因此需先找到网页源代码的编码方式并重新对其进行解编码。如图，源代码的编码方式是ISO-8859-1，对其重新解编码后其中的中文可以正常显示。



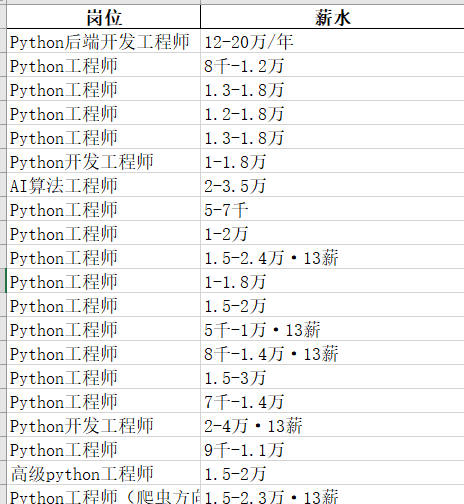
接下来我们利用re库分析软件学院首页源代码中的功能栏目。利用正则表达式list\.jsp\?urltype=tree\.TreeTempUrl&wbtreeid=\d\d\d\d">(.\*?)</a></li>可以得到网址中所有符合要求的内容。



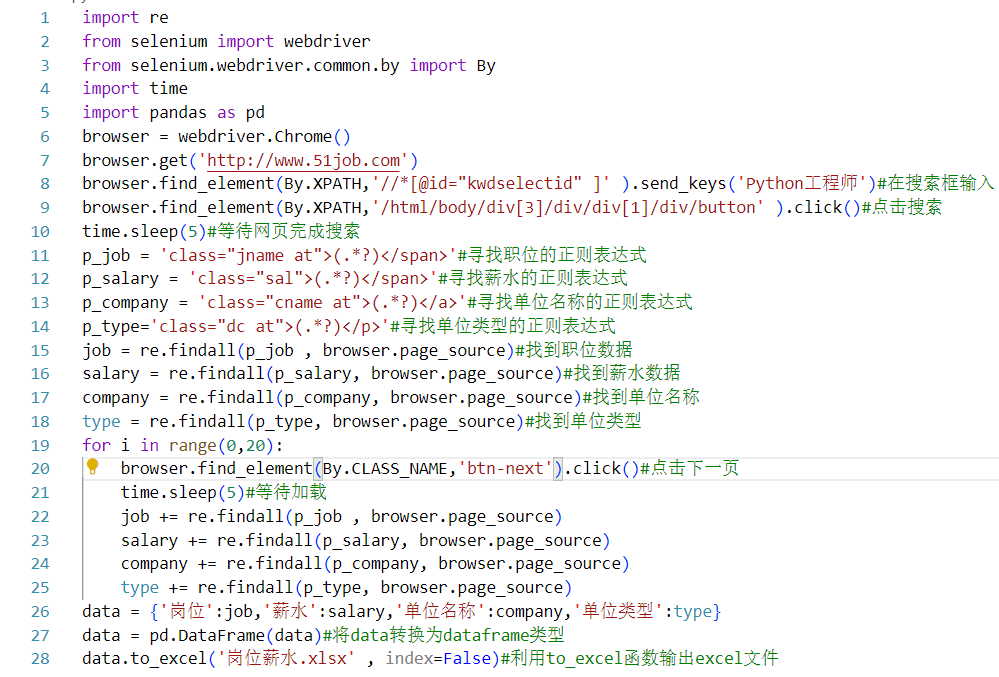
但是，requests不适用于动态变化的网页，它只能获取到网页被访问时的源代码。selenium模块可以解决这一缺陷，该模块可以在网页上模拟人的操作，如输入、点击、刷新等。selenium模块可以使用xpath定位网页元素，找到网页上的按钮、输入框后执行相关操作。page\_source属性存储了网页的源代码，可以利用re库从中获取想要的信息。pandas的DataFrame数据类型提供生成excel文件的方法to\_excel，可以加以利用。



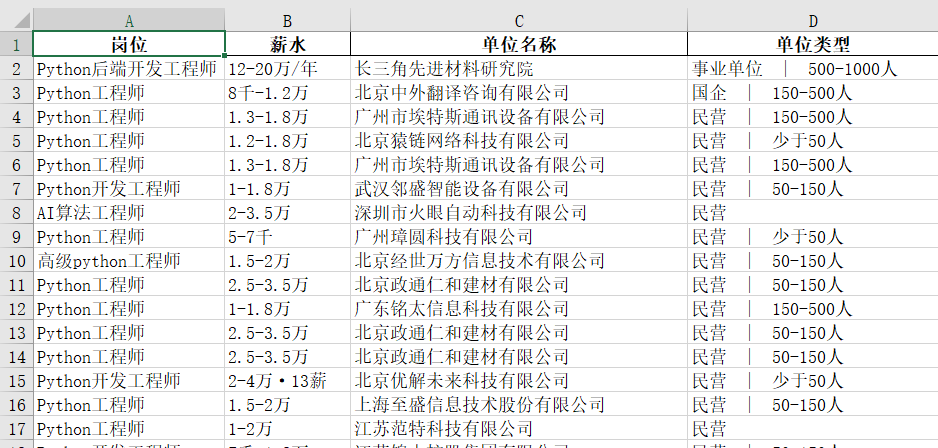
以上程序对51job网站进行爬取，获取其中Python工程师相关的职位与薪资。其运行输出一个excel文件，内容如下：



可以看到我们成功获取了网页相关内容。但以上程序爬取到的内容还不够多，改进该程序使之可以爬取单位名称、单位类型，并可以自动翻页爬取下一页数据。



实验得到的结果如下，共约420条数据：





### 实验中遇到的问题及解决

使用selenium模块时遇到问题：



这是因为电脑上的chromedriver是较久之前安装的，只适配108版本的Chrome，但Chrome已自动更新至114版本。重新下载并配置chromedriver即可解决。

将数据转换至Excel时，需要先将data转换为pandas的DataFrame数据类型，再调用其to\_excel方法。data = pd.DataFrame(data)语句报错：



即所有数组应具有相同长度。经排查，是前面的正则表达式输入有误导致re.findall没有一个job都没有找到。修改正则表达式后，data.to\_excel语句报错：



缺少必要的库openpyxl。用pip安装该库后，程序可以正常运行。

### 实验总结

实验利用requests模块爬取了软件学院官网，获得了软件学院官网的功能模块信息。实验还利用selenium模块对51job网站进行了爬取，获取了Python工程师等的相关岗位、单位名称、单位类型、薪资等。实验学习和练习了这些模块的使用，为以后可能的应用打下基础。

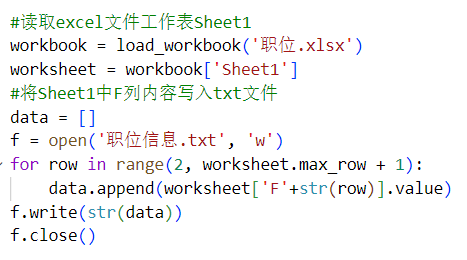
## 实验二：办公自动化

### 实验内容

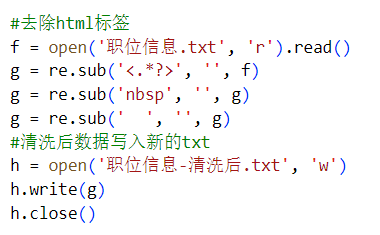
利用jieba和wordcloud等模块，根据实验一获取到的数据，进行数据清洗并生成词云图；利用opencv模块实现签名图片的处理，并利用python-docx模块自动批量生成word简历。

### 实验过程及结果

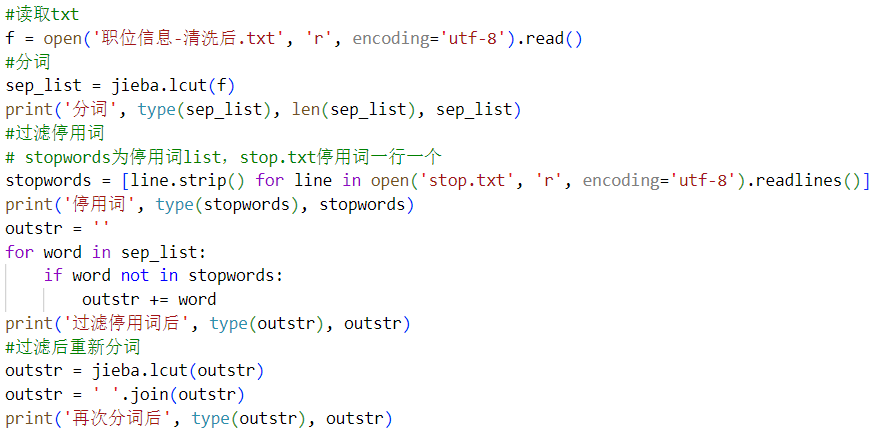
利用openpyxl的相关功能可以读取实验一中爬取到的表格文件，并将其内容写入txt文件中。

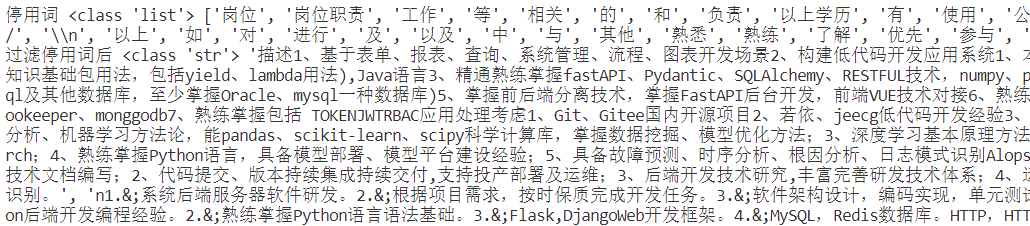


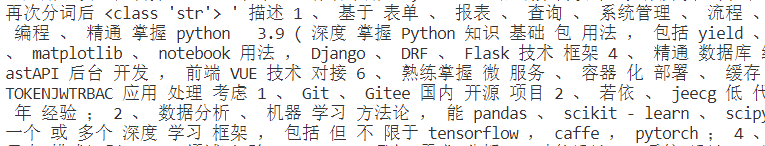
利用re库和正则表达式，我们可以对txt的内容进行数据清洗，删去其中的html标签。



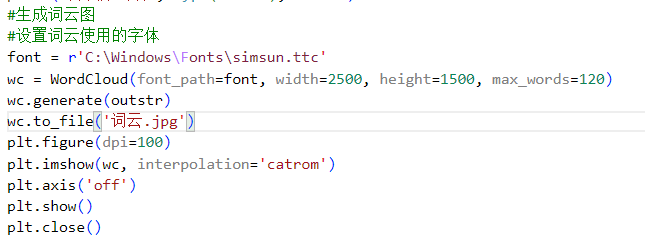
数据清洗后，利用jieba库进行词划分，并根据设置的停用词stop.txt过滤划分后的词汇，以将其中的符号、常用词过滤掉。过滤后再次进行词划分。







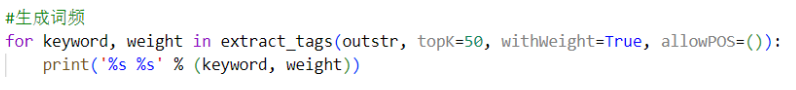
利用wordcloud模块，对以上得到的结果绘制词云图。

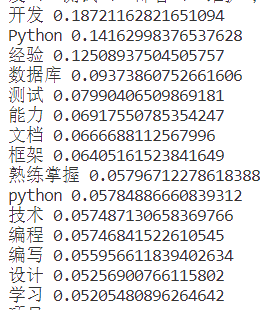


生成的词云图如下：



利用jieba.analyse的extract\_tags函数，输出各个词的词频。





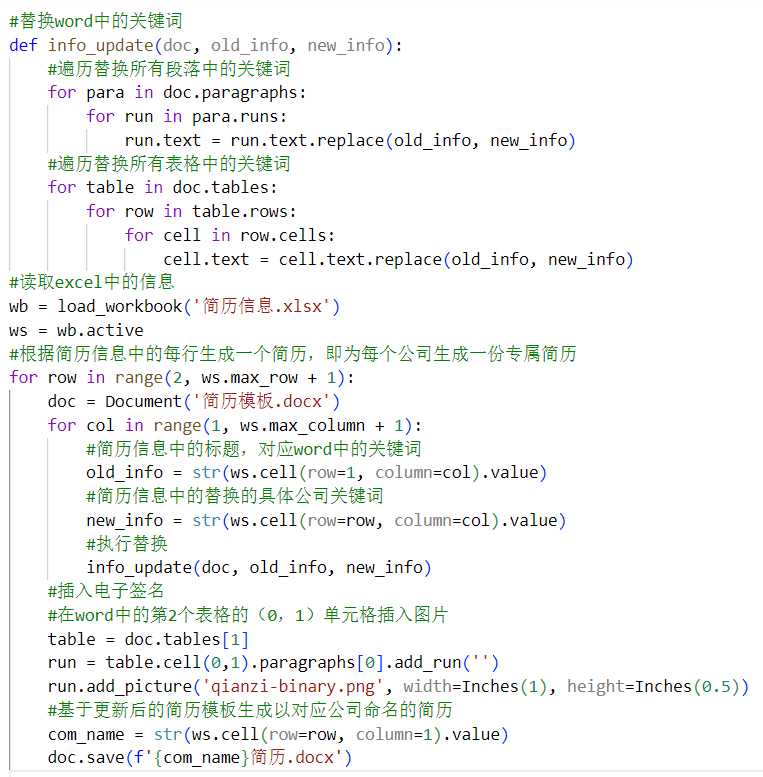
利用opencv模块可以对签名图片进行处理。

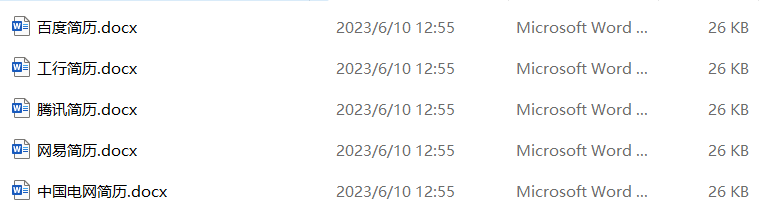


处理前的图片如下：

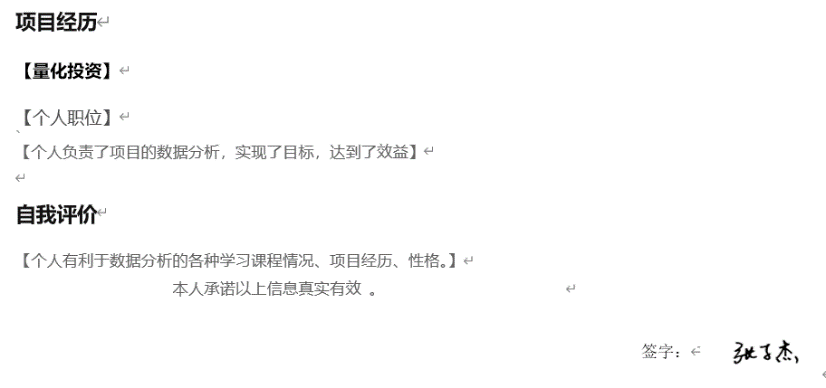
处理后的图片如下：

利用python-docx库中的replace、add\_picture等可以实现对word中文字的批量替换、图片的插入，结合循环的使用，就可以批量生成简历：









### 实验中遇到的问题及解决

绘制词云图前的过滤停用词过程报错：UnicodeDecodeError: 'gbk' codec can't decode byte 0x80 in position 33: illegal multibyte sequence。查看文件“职位信息-清洗后”发现，其编码格式为utf-8而非gbk，需要修改指定的读取编码格式。

### 实验总结

实验首先利用re和jieba模块对实验一获取的详细岗位信息进行数据清洗，之后利用wordcloud模块，生成词云图并绘制；实验还利用了opencv模块实现签名图片的二值化和透明化处理，并利用python-docx模块，向简历模板中自动插入公司、岗位、个人评价、签名等信息，批量生成word简历。实验所用到的模块和功能实用性强，这些实用小功能可以为以后的工作和学习过程提供便利。

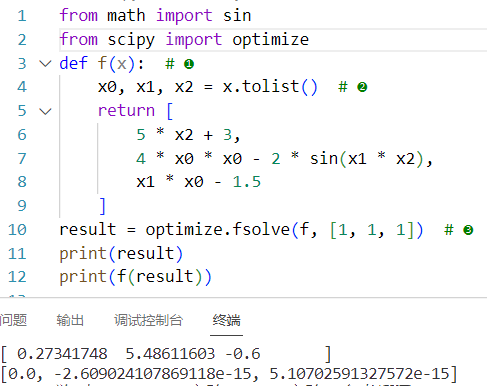
## 实验三：科学计算

### 实验内容

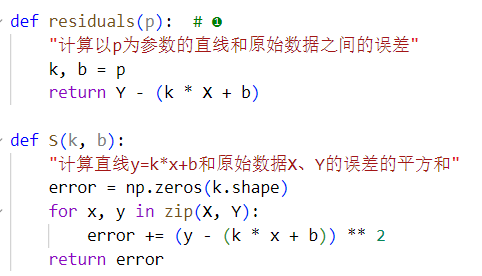
利用numpy，scipy等计算库实现非线性方程组求解、最小二乘拟合、特征值与特征向量的计算、椭圆拟合求解等，并利用dijkstra寻找图中的最短路径。

### 实验过程及结果

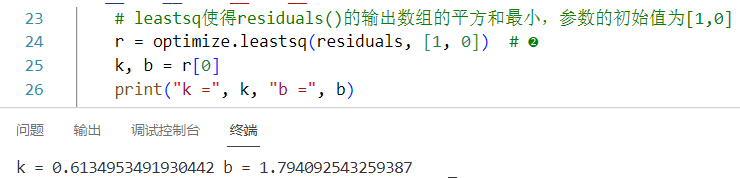
利用scipy的fsolve函数可以对非线性方程组进行求解。定义一个函数f(x0,x1,x2)，返回方程组的左侧（右侧均为0），将f放入fsolve函数求解。程序及运行结果如下所示，f(result)理论上应全部为0，但求解存在一定微小误差，误差为10-15量级。



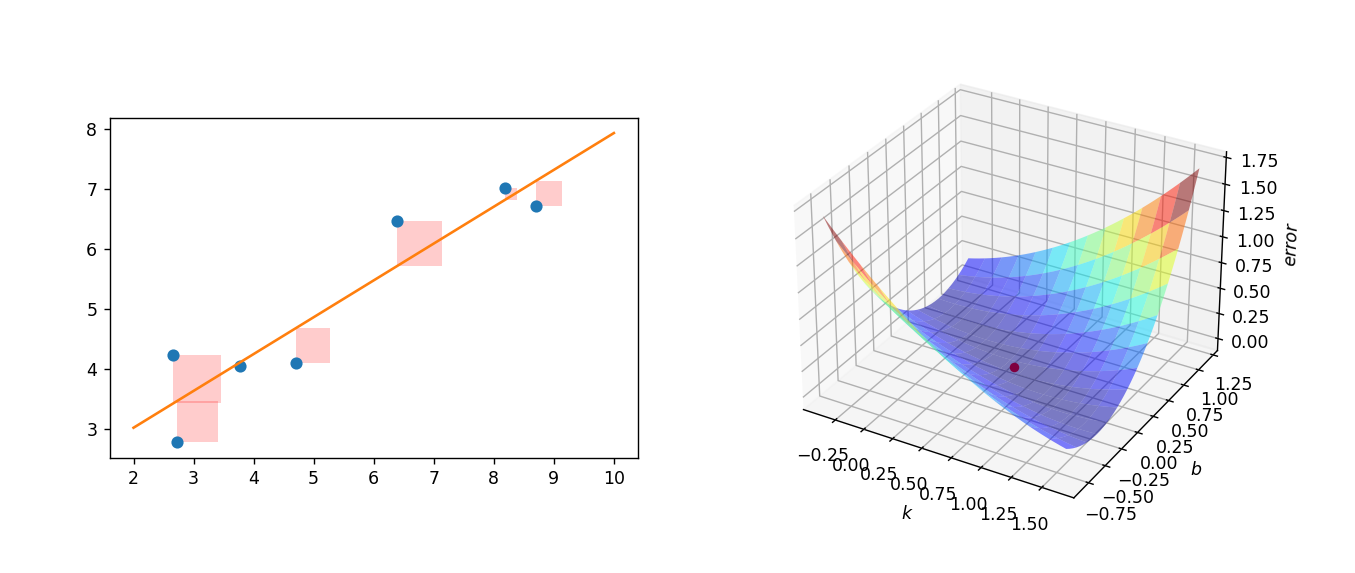
利用scipy亦可以进行最小二乘拟合。以下函数定义了误差函数，scipy的leastsq函数可以根据误差函数求取最贴近所提供数据的线性、正弦等拟合函数。



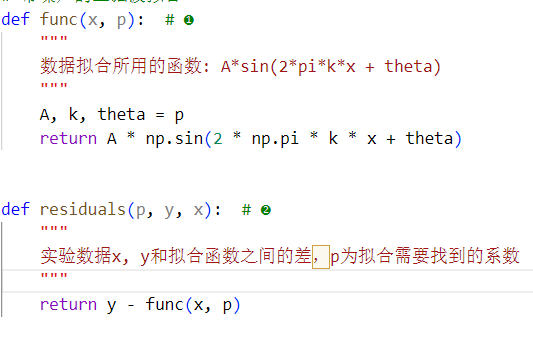
进行线性拟合：

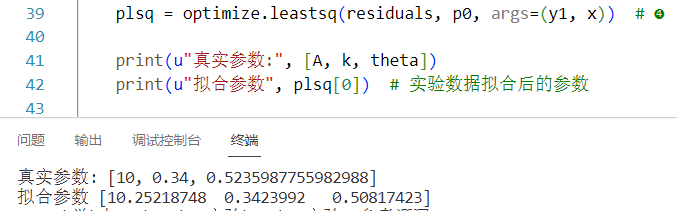


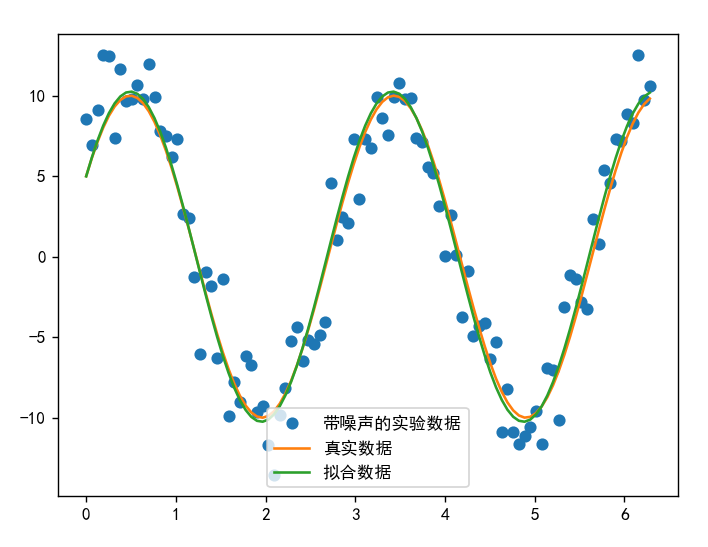
利用pylab（matplotlib的一部分，与pyplot相比更注重交互）绘制拟合结果。



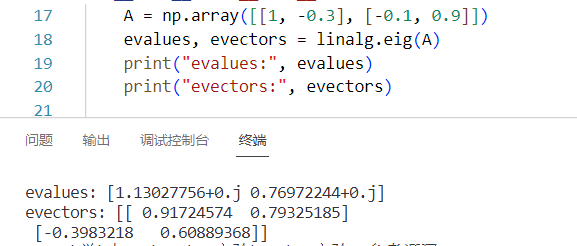
亦可以进行正弦函数的拟合。这个程序根据一个正弦函数加上噪声（随机数）生成实验数据，将实验数据利用scipy进行拟合，将实验数据、真实数据和拟合数据进行对比。

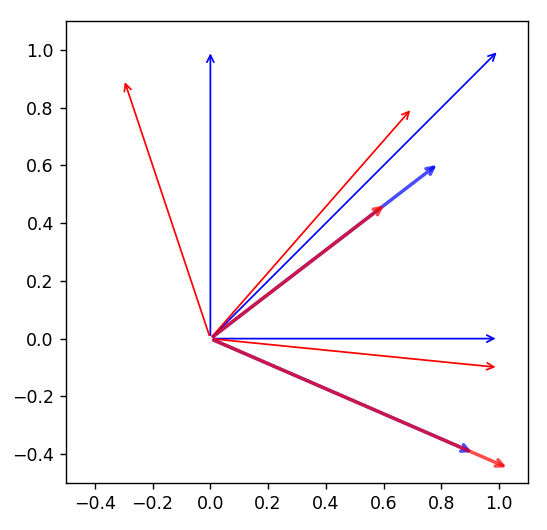




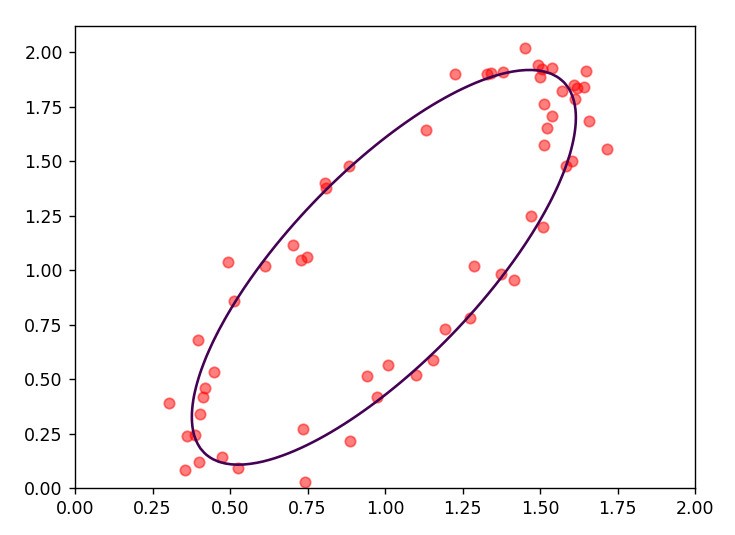


利用scipy的linalg.eig函数可以求解矩阵的特征值和特征向量。

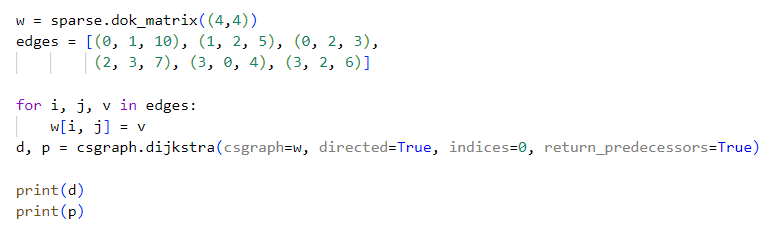


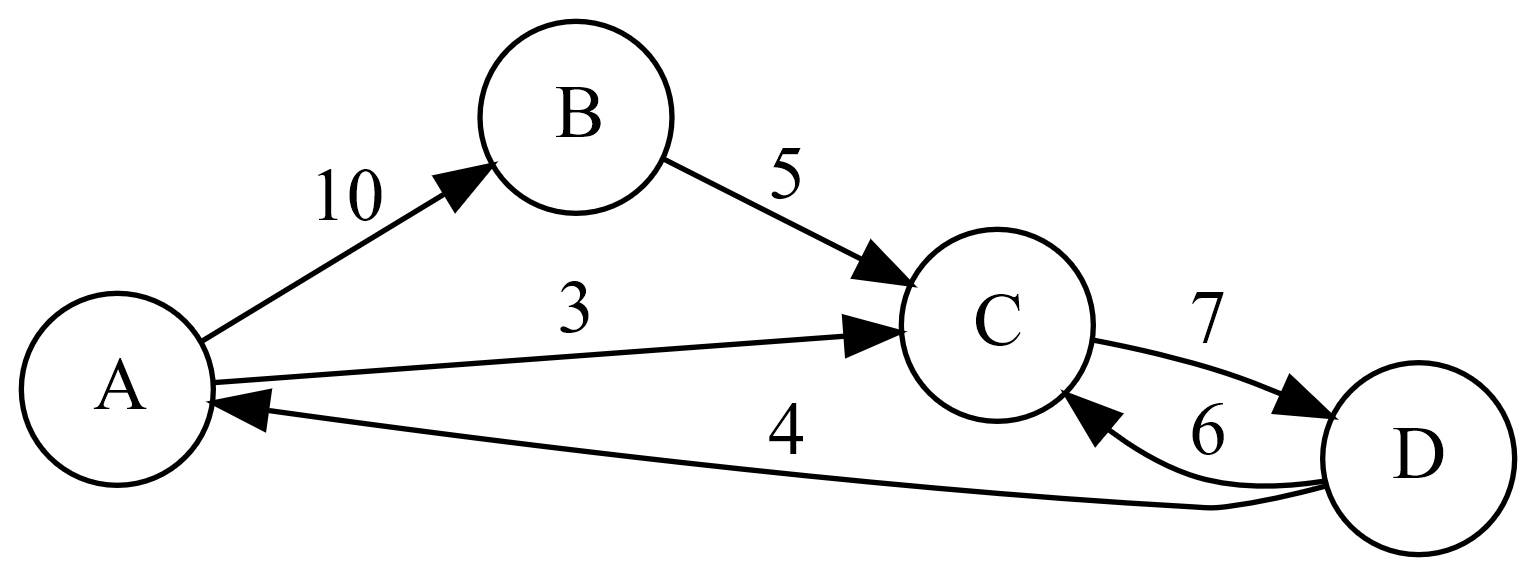


通过计算广义特征向量特征值，我们可以对数据进行椭圆拟合。

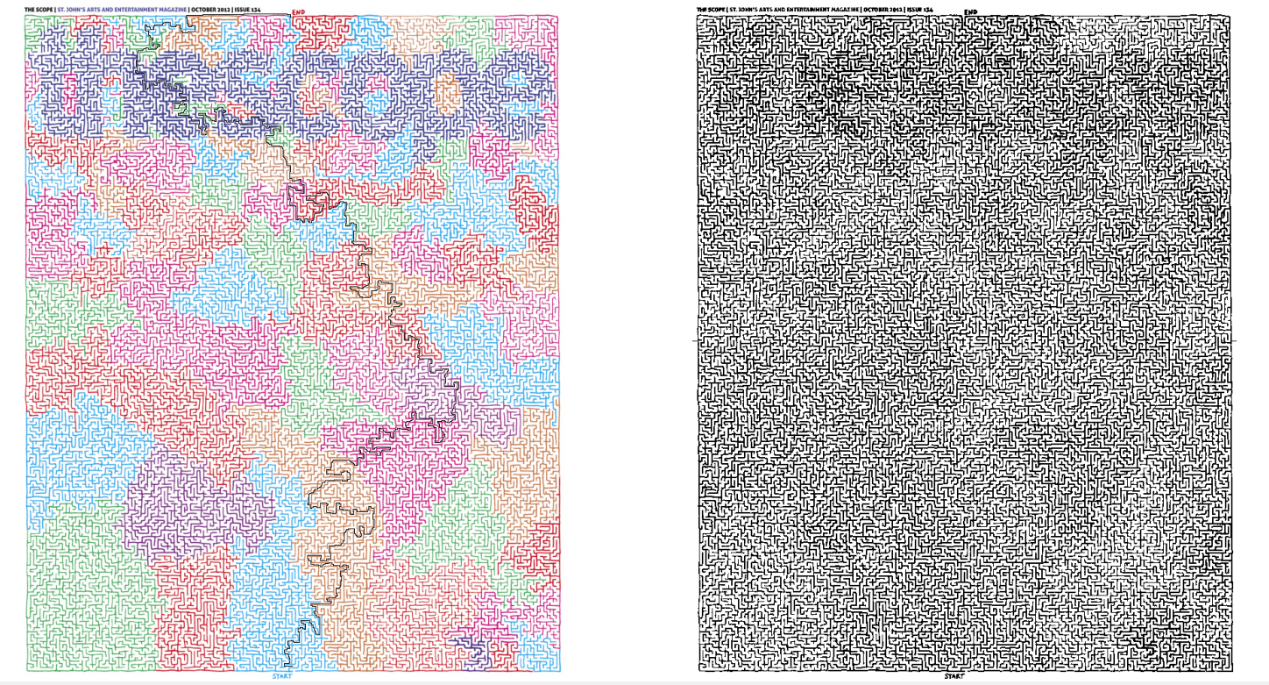


运用scipy. sparse.csgraph等可以实现利用dijkstra求解图中的最短路径。





对迷宫进行二值化处理，将其抽象为一个图结构，可以找到其中的最短路径。



### 实验中遇到的问题及解决

运用dijkstra求解最短路径的程序中报错：FileNotFoundError: [WinError 2] 系统找不到指定的文件，语句是subprocess.Popen。结合上下文注释分析其含义，推测是调用了Graphviz图可视化工具，而电脑上没有安装该工具。安装后程序可以正常运行。

### 实验总结

实验主要利用scipy库实现了非线性方程组的求解、矩阵特征值的求解、最小二乘法进行线性拟合、正弦函数拟合、椭圆拟合、寻找图中的最短路径等。该库在数值和函数计算方面具有非常强大的能力，结合matplotlib、numpy等库，可以广泛用于实验数据的分析、数学模型的求解等。

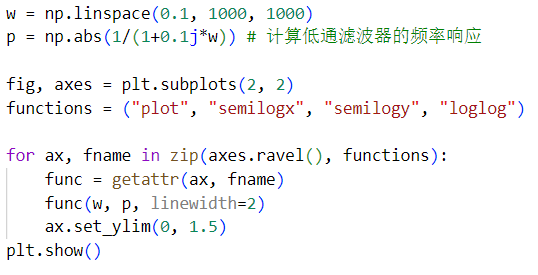
## 实验四：可视化

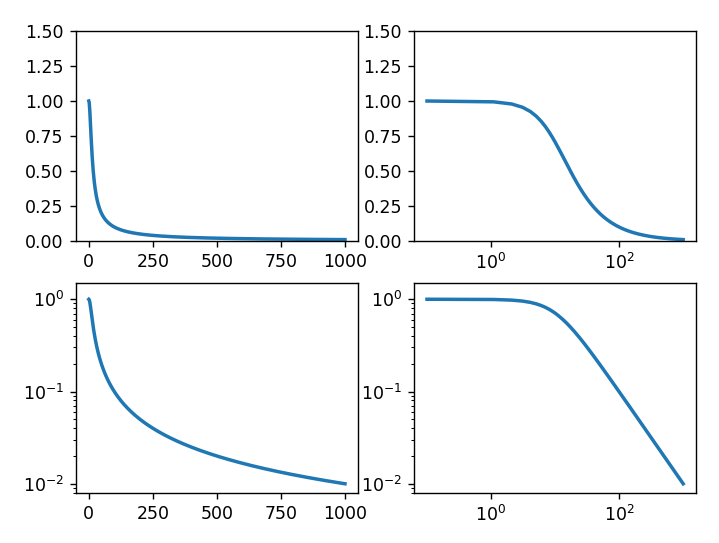
### 实验内容

本实验利用matplotlib实现数据的可视化，绘制对数坐标图、柱状图、散列图、图像、等值线、三维图等。

### 实验过程及结果

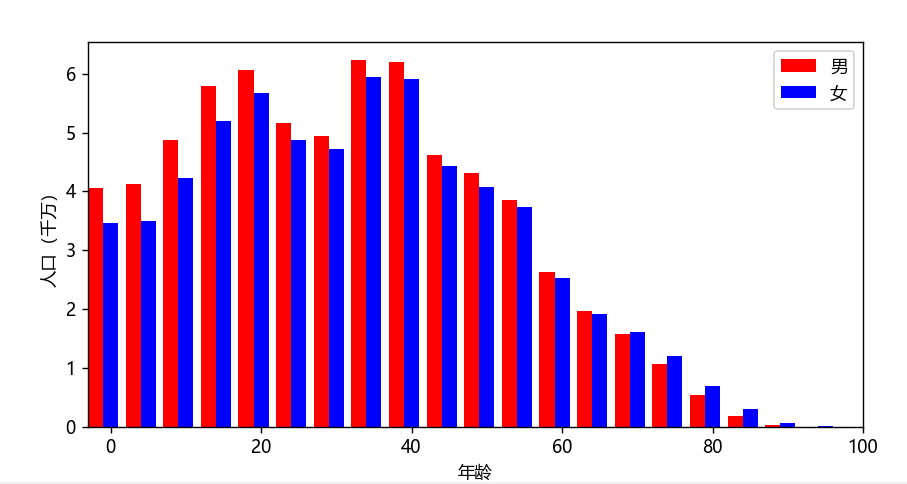
利用pyplot，绘制对数坐标图：



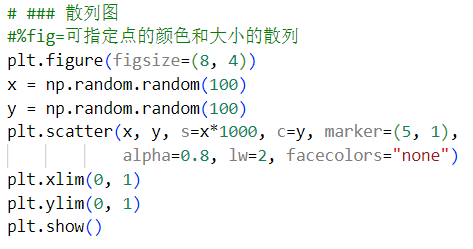
‘

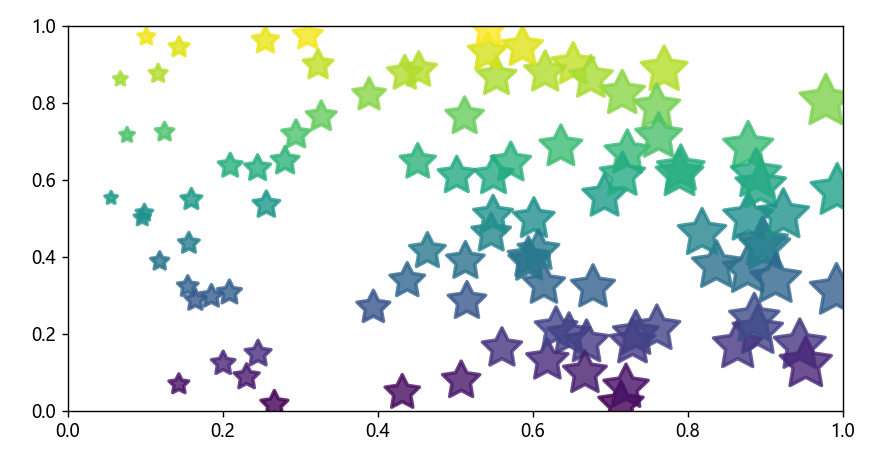
绘制柱状图：





绘制散列图：





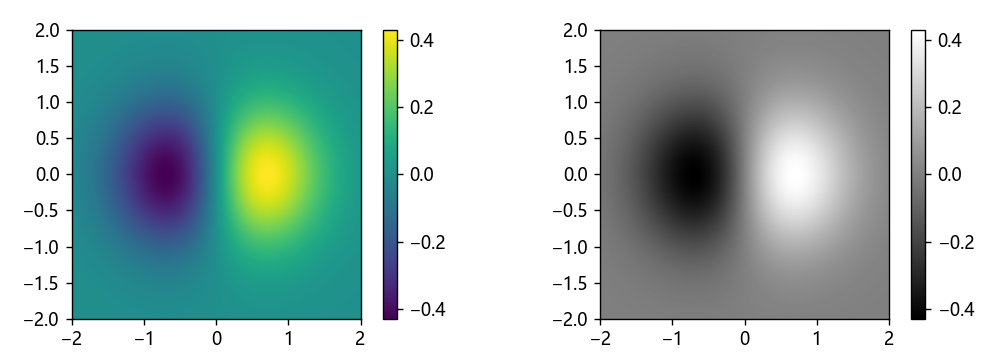
绘制图像：





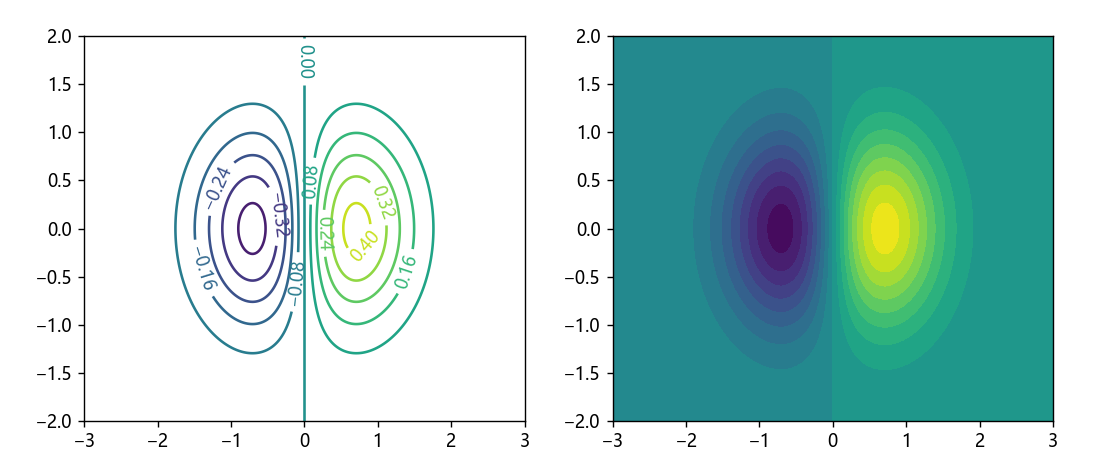
绘制可视化二元函数：

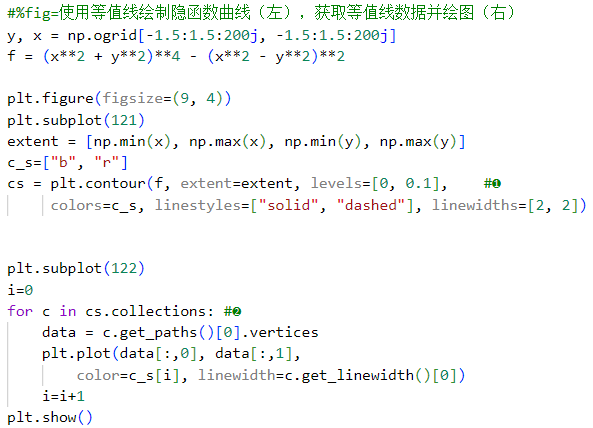


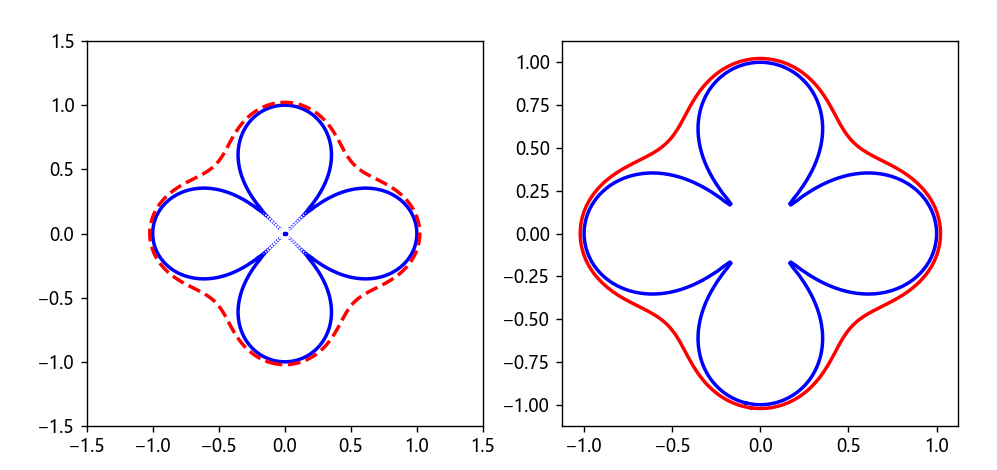


绘制等值线图：

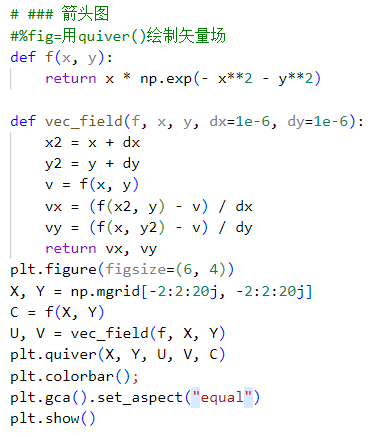


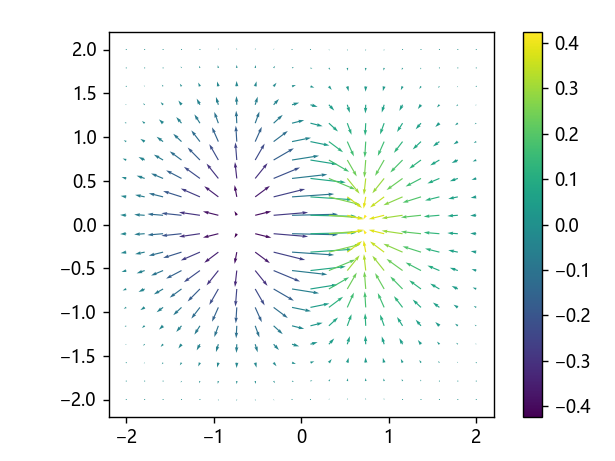




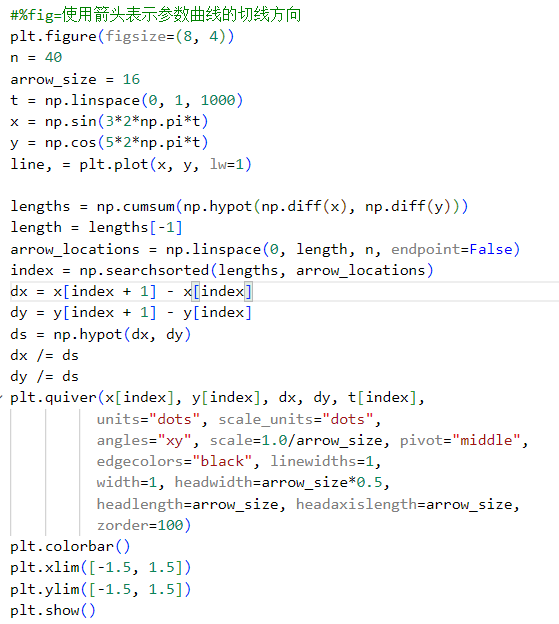


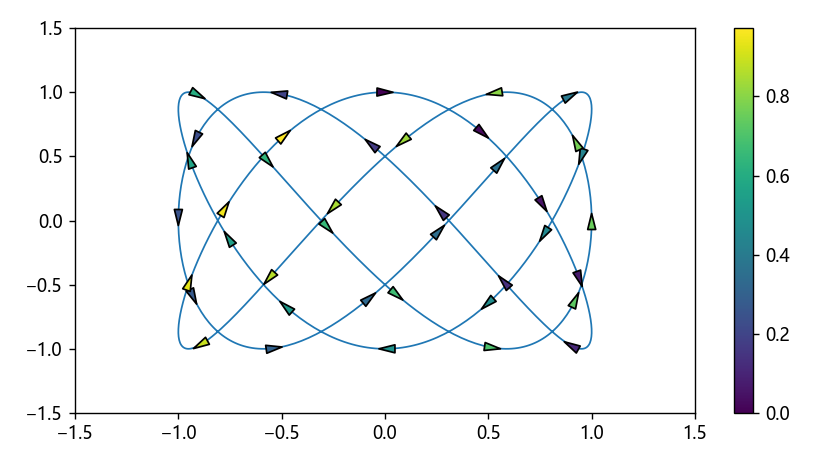
绘制箭头图（矢量场）：





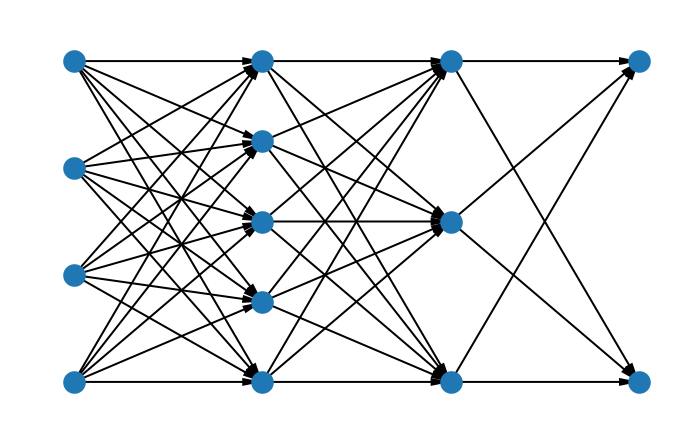
绘制箭头图（曲线的切线方向）：



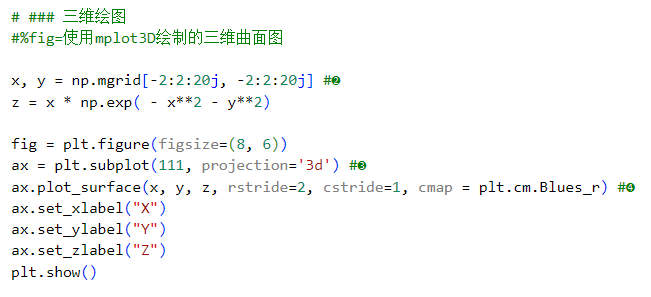


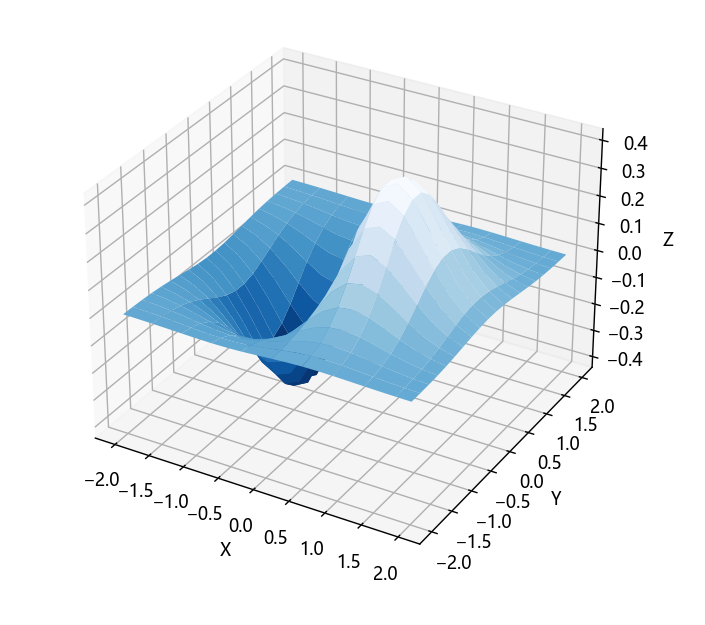
绘制箭头图（神经网络结构示意图）：





绘制三维图：





### 实验中遇到的问题及解决

绘制图像处的语句cmap\_names = list(cm.cmaps\_d.keys())报错module 'matplotlib.cm' has no attribute 'cmap\_d'。键入cm.，根据快捷提示知matplotlib.cm有cmap\_listed而没有cmap\_d，考虑是由matplotlib版本升级引起的名称变化。将语句修改为cmap\_listed，程序可以正常运行。

因不明原因，绘制图像时第四个和第五个图像显示为黑色。将代码中的255.0调整为2.0后，图像可以显示出来。

### 实验总结

本实验学习了matplotlib库在绘图方面的应用，绘制了多种多样的图表。在实际使用中，我们可以根据自己的需求选择需要绘制的图表类型，并利用python和matplotlib.pyplot进行绘制。

## 实验五：自动化选课系统（未完善）

### 实验内容

利用selenium模块，实现对选课系统的自动控制，可以实现自动刷新选修课余量并抢选。

### 实验过程及结果

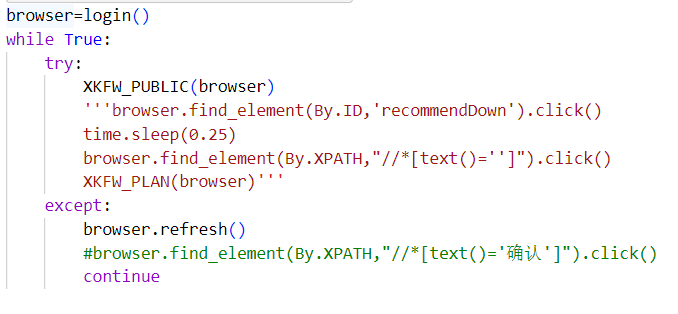
代码如下：



登录函数，登录选课系统。



这两个函数实现对选修课选课页面的自动抢选。



这是主函数，实现不断地刷新、调用抢选函数。

### 实验总结

该程序为2022年2月、8月所编写，运用selenium实现了自动控制。目前选课系统未开放，无法对程序进行调试，其是否还能使用也未可知。程序还希望实现对主修课的抢选功能，但该段功能运行有bug且未修复。总体上该程序是一个半成品，待下次选课期间继续完善。