**（封面）**

**《Python数据分析》**

**课程综合作业报告**

此处写明题目

**(题目字号小三，中文宋体，英文和数字是Times New Roman)**

专 业

学生姓名

学 号

所属班级

2022年 6月 1日

(统一写上面的时间)

影视作品分析

1. 设计背景和目的（一级小标题黑体，小四号字）

随着互联网技术越来越发达，人们网上活动越来越频繁。影视评论也越来越丰富。本文通过清洗处理分析夏洛特烦恼电影评论数据集，分析该电影评分分布以及评论人员地区分布，最后通过词云分析评论词频即可得知该电影主要特点以及风评。

（二级、三级等小标题，汉字是宋体，英文和数字是Times New Roman，小四号，1.25倍行距，）

正文部分字体，汉字是宋体，英文和数字是Times New Roman，小四号，1.25倍行距，首行空两个字符（空两个汉字）。

二、相关技术

数据读取处理技术 pandas

数据可视化技术 matplotlib

词云技术 wordcloud

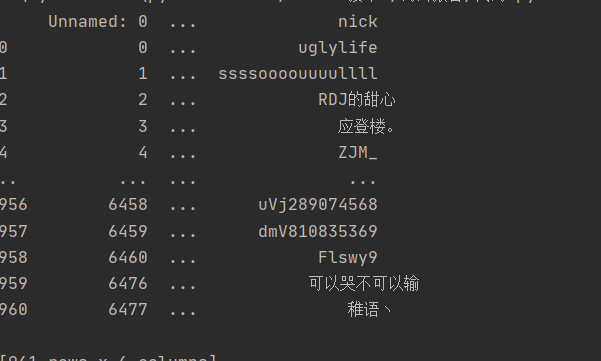
分词技术 jieba

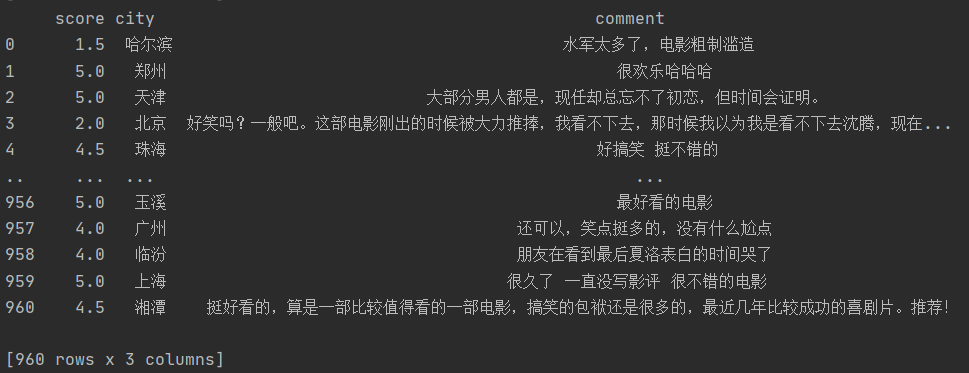
三、总体设计

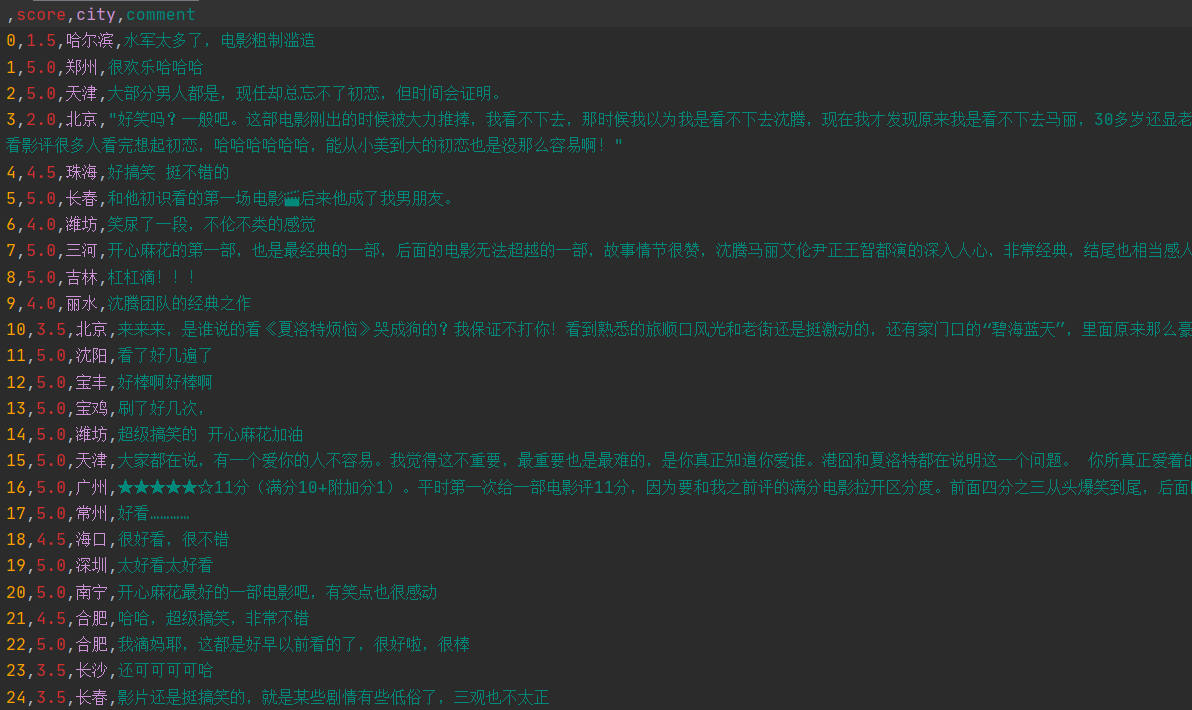
开发环境使用python3.8，编译器使用pycharm，通过pandas读取Excel文件中的数据以及初步清洗。清洗完毕后，转化为列表。通过列表推导式等操作提取画图分析所需要的xy轴列表。通过matplotlib库画图，最后通过jieba和wordcloud展示词云图。

1. 详细设计
2. 通过pandas导入数据，提取出对我们有作用的列，去除掉有空值的行，保存为新的处理后的数据。

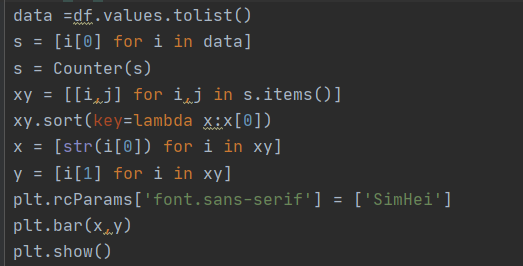


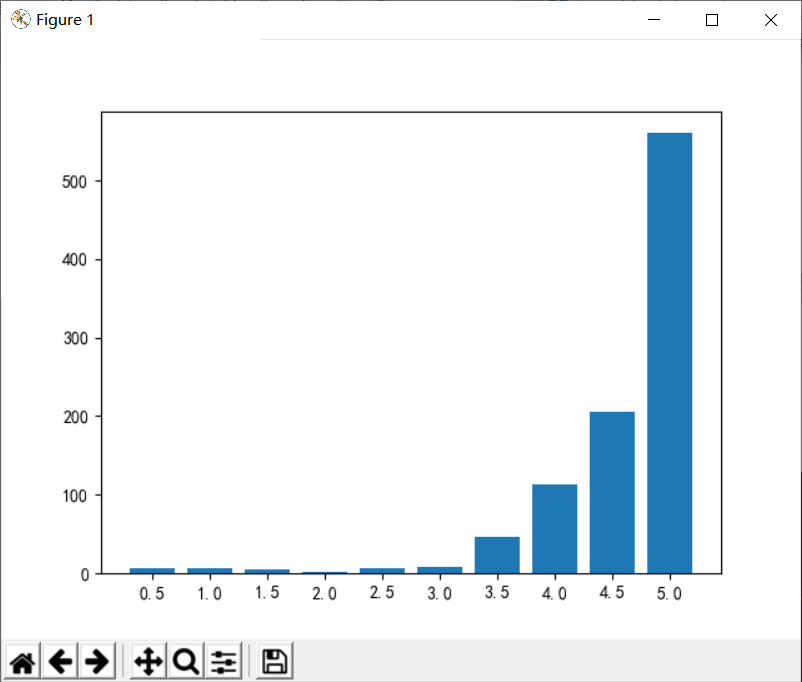




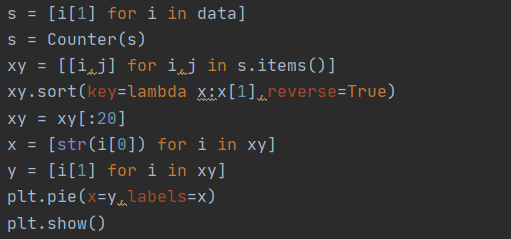


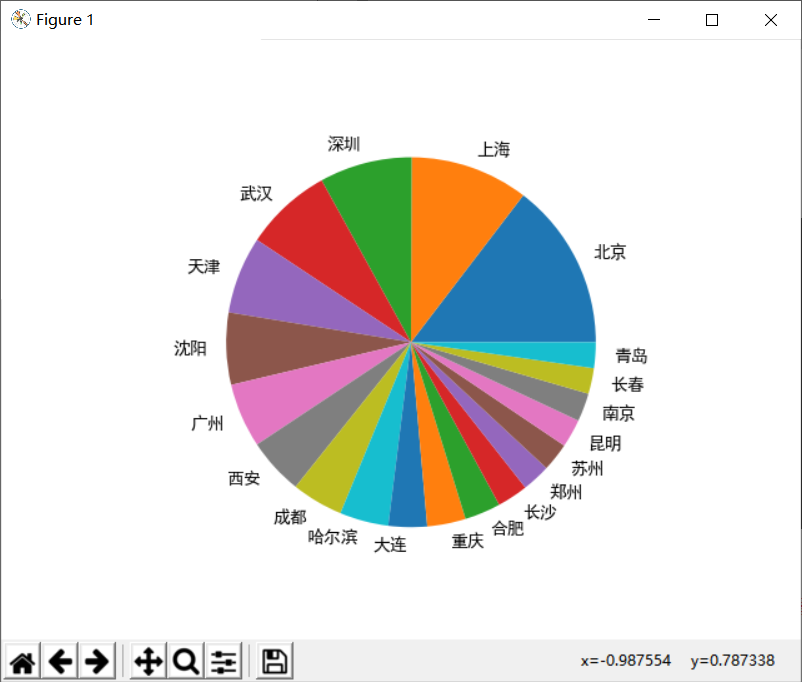
1. 将数据转为列表以后，通过提取数据的第一列也就是评分列作为子列表，对子列表调用Counter统计不同评分出现频率获得字典，之后通过字典的键值对构造新列表作为xy轴列表进行画图。





1. 同理，利用上述相同的操作，将第一列改为第二列城市列以后提取xy轴进行画图，由于城市数量过大，排序以后只提取数量最大的前20个城市进行画图。





1. 数据分析基本完毕以后单独提取第三列评论文本列，引入停用词库文件，之后利用jieba分词对文本进行分词，将存在于停用词库内的词去除，构造词语列表。同样利用Counter统计词频字典以后，利用Imgae读取背景图片二进制数据，传入wordcloud generation生成器中构造词云图。

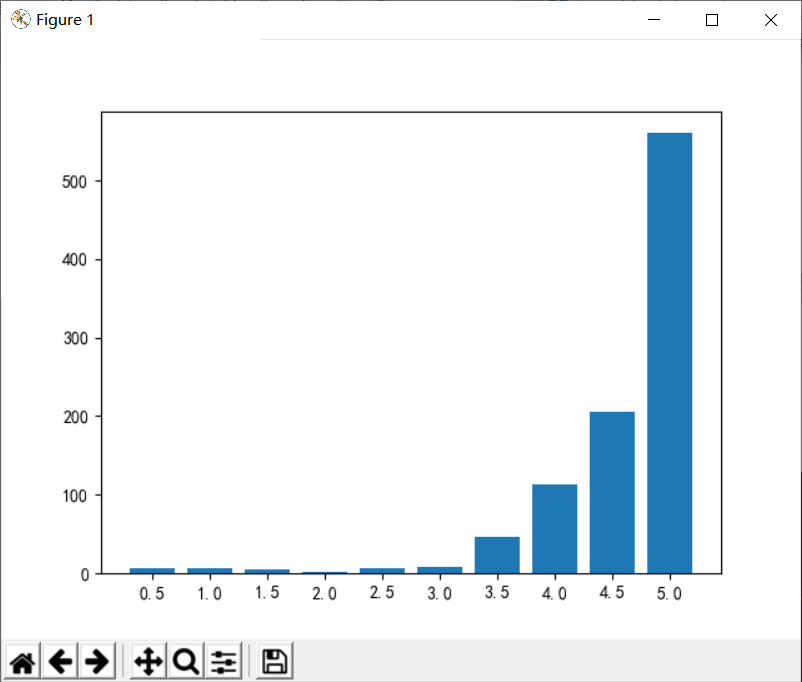




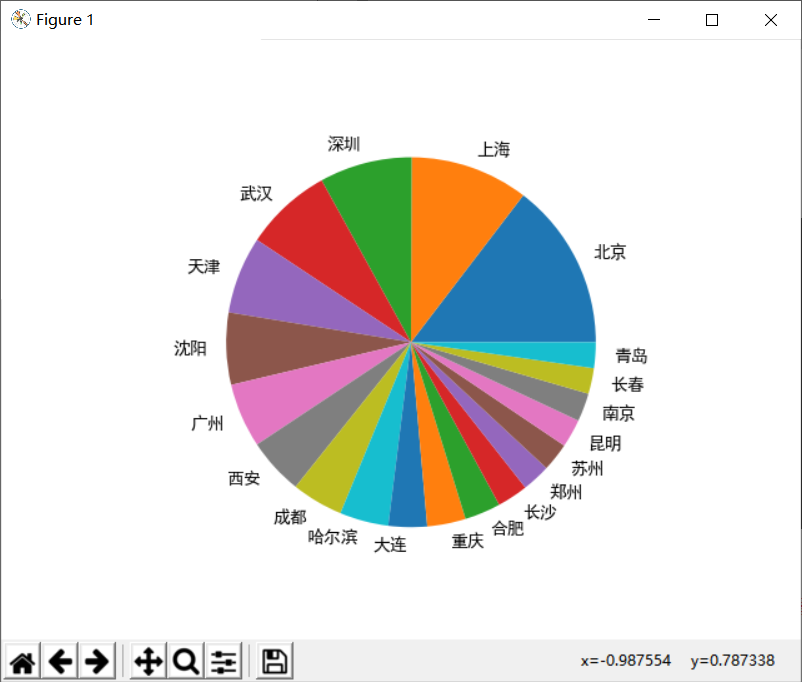
（分模块完成：如数据获取、数据探索、数据处理、数据分析等）

五、分析结果和结论：

通过评分分布柱状图可以看出来这部电影的平均评分还是比较高的，主要集中在4.5分与5分，可见大众对这部电影的评价还是比较高的。



通过城市分析饼图可以看出大部分的评论数据主要还是集中在一线城市，新一线城市以及二线城市，可以看出互联网活跃度与城市人口和城市发展存在一定的联系。



最后通过网友评论的文本词云图可以明显看出这部电影是一部喜剧风格明显的电影。评价词大部分也比较积极正面，值得一看。



六、总结

通过本项目分析了夏洛特烦恼的电影评论数据以及该电影的整体情况。同时学会了许多python的基础知识以及数据分析可视化的相关扩展包的使用。

本项目的结论和得到的启示、收获。

1. 源程序清单和执行结果：

import pandas as pd  
import matplotlib.pyplot as plt  
import wordcloud  
from collections import Counter  
import jieba  
import numpy  
from PIL import Image  
df = pd.read\_excel('夏洛特烦恼.xlsx')  
print(df)  
df = df[['score','city','comment']]  
df.dropna(axis=0, how='any',inplace=True)  
df.to\_csv('清洗后数据.csv',encoding='utf-8')  
print(df)  
data =df.values.tolist()  
s = [i[0] for i in data]  
s = Counter(s)  
xy = [[i,j] for i,j in s.items()]  
xy.sort(key=lambda x:x[0])  
x = [str(i[0]) for i in xy]  
y = [i[1] for i in xy]  
plt.rcParams['font.sans-serif'] = ['SimHei']  
plt.bar(x,y)  
plt.show()  
s = [i[1] for i in data]  
s = Counter(s)  
xy = [[i,j] for i,j in s.items()]  
xy.sort(key=lambda x:x[1],reverse=True)  
xy = xy[:20]  
x = [str(i[0]) for i in xy]  
y = [i[1] for i in xy]  
plt.pie(x=y,labels=x)  
plt.show()  
s = [i[2] for i in data]  
object\_list = []  
with open('停用词库.txt', 'r', encoding='UTF-8') as meaninglessFile:  
 stopwords = set(meaninglessFile.read().split('\n'))  
stopwords.add(' ')  
for i in s:  
 #对每一行内容，进行jieba分词处理，模式选择精确分词模式  
 seg\_list\_exact = jieba.cut(i, cut\_all=False, HMM=True)  
 #打开停用词文件，将分词成果去掉所有的停用词  
 for word in seg\_list\_exact:  
 if word not in stopwords:  
 object\_list.append(word)  
# 统计词语出现频率  
object\_list = Counter(object\_list)  
# 选择词云背景图  
mask = numpy.array(Image.open('heart.png'))  
# 设置词云参数  
wc = wordcloud.WordCloud(  
 font\_path = 'C:/Windows/Fonts/simfang.ttf',  
 background\_color='white',  
 mask = mask,  
 max\_words = 500,  
 max\_font\_size = 120  
)  
# 生成词云对象  
wc.generate\_from\_frequencies(Counter(object\_list))  
wc.recolor(color\_func=wordcloud.ImageColorGenerator(mask))  
plt.figure('词云')  
plt.imshow(wc, cmap=plt.cm.gray, interpolation='bilinear')  
# 显示词云  
plt.show()

此部分是源代码

注意：红色字部分要求删除，换成需要的内容。