Python项目实训报告

|  |  |
| --- | --- |
| 系别 |  |
| 班级 |  |
| 项目名称 |  |
| 成绩 |  |

年 月 日

目 录

[一、 项目需求（开发主题，目的） 2](#_Toc17373)

[二、 项目设计分析（根据总体设计） 2](#_Toc29416)

[三、 程序设计 2](#_Toc17361)

[四、 项目测试（所有功能测试截图） 2](#_Toc15651)

[五、 相关技术说明（写出核心技术及代码说明） 3](#_Toc14968)

[六、 Bug及不足（写出存在Bug及项目不足） 3](#_Toc15946)

七、 实训总结 3

# 项目需求（开发主题，目的）

# 目前在处于大数据时代，对于消息不仅要在表格上表示出来，更需要通过图表更直观的展示消息，数量和类型等之间的关系，所以，所以此项目用python做数据的可视化分析，这样更能直观地展示出各个阶段，因变量的变化，从而推断出未来的发展趋势和人们的喜好类型

# 项目设计分析（根据需求分析设计）

1. 主程序设计
2. 提取数据
3. 数据清洗
4. 数据分类
5. 数据不同组合
6. 数据的挖掘
7. 函数设计
8. 绘制柱状图函数：
9. 绘制线性函数
10. 绘制饼状图
11. 绘制直方图
12. 绘制世界分布图
13. 生成导演，演员人物词云

# 程序设计

提取数据

df = pd.read\_excel("1.xlsx", sheet\_name=0)  
title = df['标题']  
  
dir = df['导演: ']  
scip = df['编剧:']  
actor = df['主演: ']  
  
type = df['类型']  
country = df['制片国家/地区:']  
year = df[' 上映日期']  
long = df['片长: ']  
type = df['类型']

数据清洗

year\_l = []  
for j,i in enumerate(year):  
 if isinstance(i, datetime.date):  
 year\_l.append(i.year)  
 else:  
 year\_l.append(i)  
year\_l.sort()

绘制柱状图函数

def max\_plot\_draw\_bar(x,y,name):  
  
 plt.figure(figsize=(120, 40)) # 定义坐标图尺寸  
 plt.title(name, fontsize=50)  
  
 # 45为旋转的角度，不然x轴会特别密  
 plt.xticks(rotation=45)  
 # 绘制柱状图  
 z = [i for i in range(len(x))]  
 plt.xticks(z, x)  
 plt.tick\_params(labelsize=50) # 坐标轴单位大小  
 for ii in y:  
 plt.scatter(z,ii)  
 savename = './pic/' + name  
 plt.savefig(savename, # province⽂件名：png、jpg、pdf  
 dpi=100, # 保存圖⽚像素密度  
 facecolor='white', # 視圖與邊界之間顏⾊設置  
 edgecolor='lightgreen', # 視圖邊界顏⾊設置  
 bbox\_inches='tight') # 保存圖⽚完整  
 print('over')  
 plt.clf()

绘制直方图

def plot\_draw\_zhifang\_bar(yy,xx,name,tt):  
  
 plt.figure(figsize=(150, 40)) # 定义坐标图尺寸  
  
 #colors = ["#8dd3c7", "#bebada"]  
 labels = tt  
 z = [i for i in range(len(yy))]  
 print(xx)  
 print('^^^^^^^^^')  
 plt.xticks(z, yy)  
 plt.tick\_params(labelsize=40) # 坐标轴单位大小  
 plt.hist(xx, bins=z,histtype="bar", rwidth=0.8,  
 stacked=True, label=labels)  
 plt.xlabel("国家", fontsize=50)  
 plt.ylabel("数量", fontsize=50)  
 plt.title(name, fontsize=50)  
 plt.rcParams.update({'font.size': 20})  
 plt.legend(labels=tt, loc='best')  
  
 plt.savefig('./pic/' +name)  
 plt.clf()

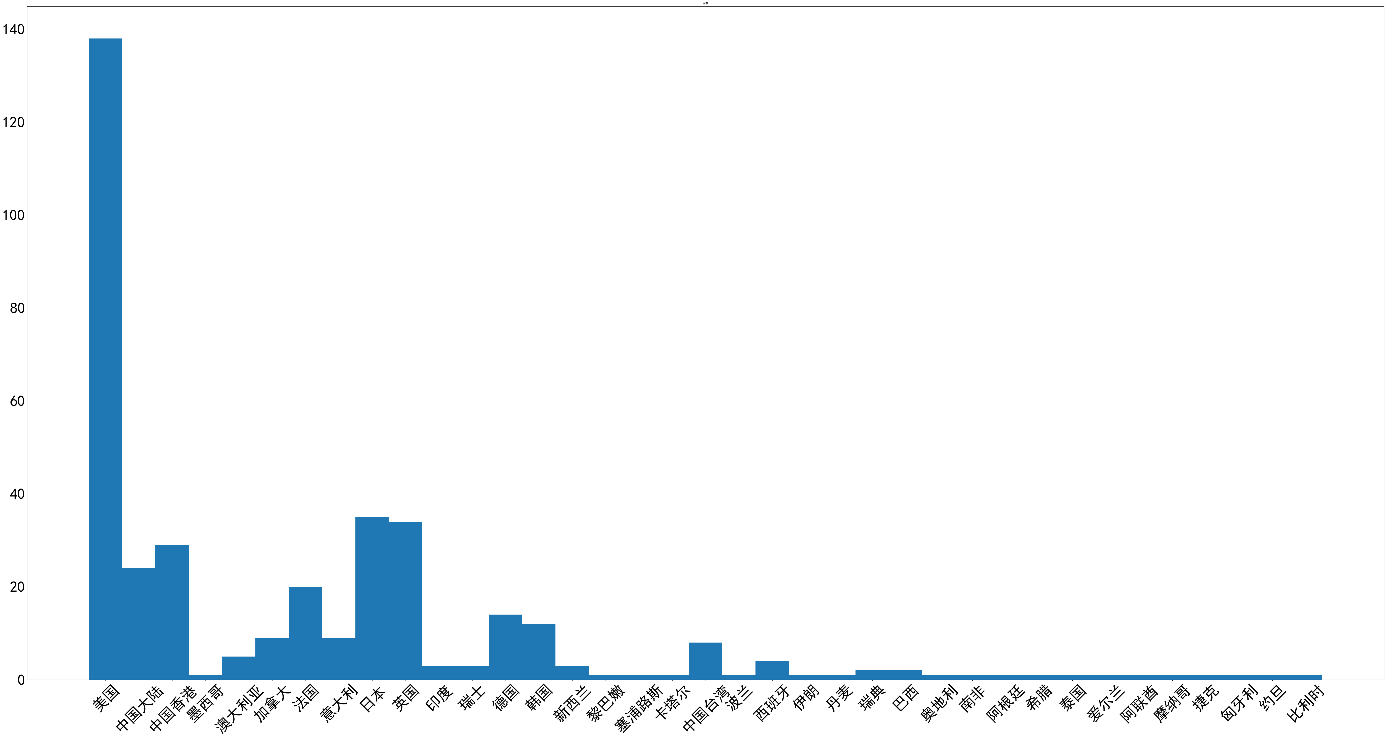
绘制世界数量分布地图

def map\_visualmap(sequence, year):  
 name\_map = {  
   
 }  
 c = (  
 Map(opts.InitOpts(width='1500px',height='800px'))  
 .add(year, [list([i,j]) for i,j in sequence.items()], "world", name\_map=name\_map)  
 .set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=False))  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title="Map-世界地图"),  
 visualmap\_opts=opts.VisualMapOpts(max\_=200),  
 )  
 .render("map\_world.html")  
 )

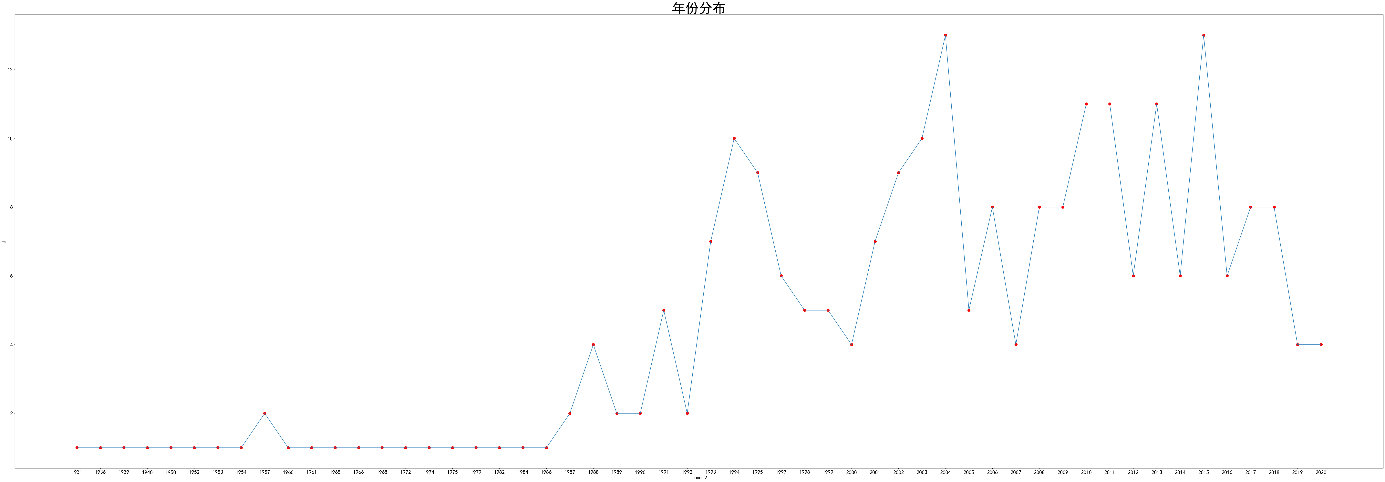
绘制人物词云图

# 项目测试（所有功能测试截图）

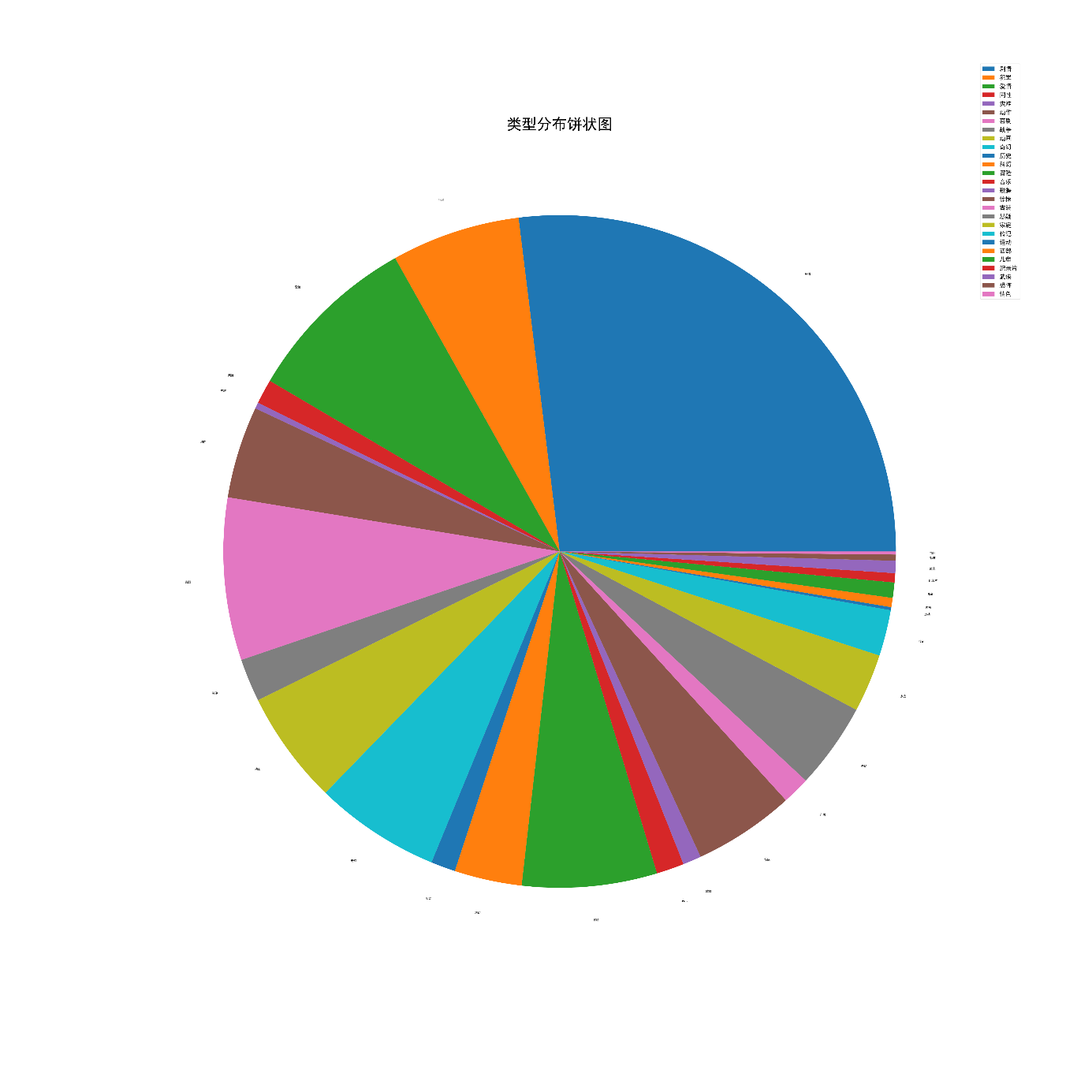
绘制柱状图函数：



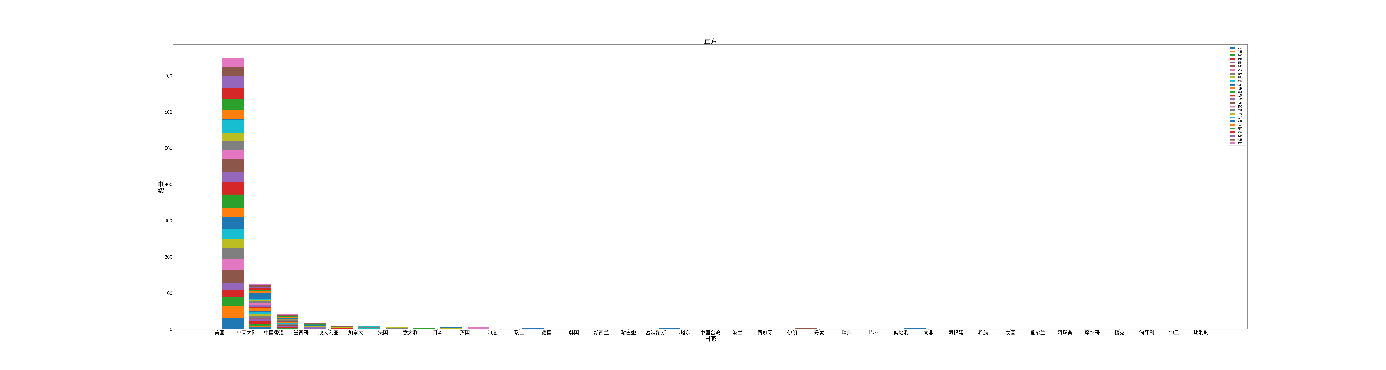
绘制线性函数



绘制饼状图



绘制直方图



绘制世界分布图

绘制人物词云



# Bug及不足（写出存在Bug及项目不足）

1. **存在Bug原因及解决思路：**

在编写程序过程中有的bug有如下几种，

1 忽略变量与变量之间的深拷贝和浅拷贝，用一个上个变量赋值给下个变量，只想改变下个变量，却把上个变量给改变了，这是Python的特点，要牢记，如果只想改变后者变量，那可以使用.copy。

2 字符的用法要注意，提取数据的过程中数据总是提取不到，结果发现是因为少了一个空格，所以说这种空格细微的差别要好好的察觉到

3 变量的引用规范，开始也用了一个x的变量，后来又在循环中又有x的变量，这使得x覆盖了上一个x的含义，总结参数一定要有他的特定名字，除非在循环里面

4 在做世界地图的时候，中国显示的电影数量一直为0，后来才发现我们选了字典键是‘中国大陆’4个字，但是这个pychartmap没有中国大陆，只能识别中国，所以说要把中国大陆改成中国，总结使用第三方库时，一定要查看手册。

5 在做年份提取的时候，比如1998年1月12日，我们只提取前面的1998在提取单个的时候没问题，但是批量提取时总是出问题，结果发现是数据不够干净，在年份里面总有一些没有年份的或者年份类型是整形，所以需要我们做清晰

1. **项目不足或有待完善细节说明**

**1** 函数的复用可以增强，比如画的各种类型的图在其他代码上也能跑，这样可以大大减小代码的可重复率,这一点需要有害代码

2 代码变量的命名规范也需要注意，最起码每个变量自己要知道什么意思而不是代码写多了自己都不知道这个变量什么意思，然后造成不必要的混淆

3 对于调试debug功能的使用不够熟练，如果熟练的话可以大大减少找bug的时间增强效率

4 绘制的图存在美观上一定美观的缺陷，可以进行优化，让图形变得更加的富有美感

1. 后期可以通过以获得的数据来做未来的预测，这也是一个可以深入的地方

**技术文档及参考资料：**

plt.pie()参数解释[https://blog.csdn.net/mariodf/article/details/121303339](https://blog.csdn.net/mariodf/article/details/121303339" \t "_blank)

pychartmap<https://blog.csdn.net/weixin_42488050/article/details/112011141>

1. **实训总结**

**通过本次python的训练，学到了很多实践的知识，学会了怎么去调试程序，怎么去规划程序，然后遇到问题怎么解决，学会善于查阅手册，问题虽然不能说，全部能解决，但是善于去寻找问题这种方法也是非常重要的，其次对拍上的各种类型，比如说字典列表字符串等，有了更深一步的理解，也懂得了他们之间存在的意义与优势，也更加体会到Python和其他编程语言的不同之处和python独特的美感，他的简洁度，灵活度和拓展度，更加了解了python在数据处理领域的强大能力。**

**本次的实践课程是非常有意义的，锻炼了动手能力，思维能力与解决问题的探索能力，也增加了我对python的热爱与兴趣**