南昌大学实验报告

学生姓名：丁俊　学　号：8003119100　专业班级：信息安全 193 班
实验类型：□ 验证■ 综合 □ 设计 □ 创新　实验日期：2021.11.21　实验成绩：

一、实验项目名称

数据爬取与分析

二、实验目的

1. 掌握 request、BeautifulSoup 和 re 等爬虫实现工具
2. 掌握 Numpy、Pandas 和 Matplotlib 基本使用
3. 掌握表示、清洗、统计和展示数据的能力

三、实验任务

1、参照《青春有你 2》、《乘风破浪的姐姐》、《安家》数据爬取与分析，自选一个主题完成数据爬取与分析。

2、参照“数据可视化 - 中国的天气数据”掌握 Basemap、seaborn、pyecharts 等可视化库的使用。（可选）

四、主要仪器设备及耗材

软件：Anaconda 或者 pycharm 等

五、实验步骤与结果

1、爬虫目标

- 猫眼电影榜单 Top100 榜，网址为 https://www.maoyan.com/board/4?offset=0

从网页中提取出 top100 的排名、电影名称、评分、上映时间、详情链接、上映地区、演员、剧照图片等。

- 根据爬虫结果，进行简单的可视化分析。

2、爬取步骤

- 网址 url 分析

首先，打开猫眼 top100 榜单电影的地址 https://www.maoyan.com/board/4?offset=0
页面非常简单，页面底部有分页显示栏目，我们要做得就是进行分页爬取，总共需要爬取 10 页数据，每一页的 html 样式和结构都类似，这里使用一个偏移量 offset 对构造 10 页地址，分别进行访问。

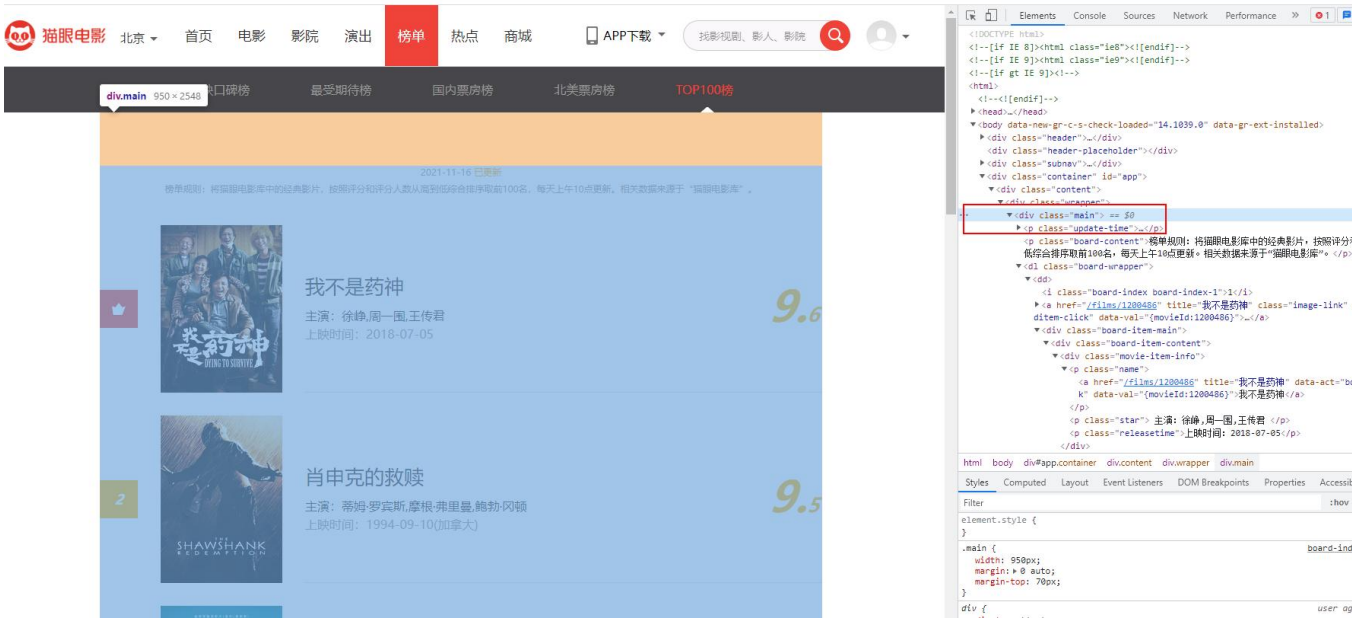
- **Requests 进行访问**

```
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 ',
}

proxy = {
    'http':'114.106.171.160:13456',
    'http':'114.104.121.169:13456',
    'http':'113.124.92.234:13456'
}
# 爬取网页并返回网页内容div块中的内容
def crawl_data(url):
    try:
        response = requests.get(url, headers = headers)
        if response.status_code == 200:
            soup = BeautifulSoup(response.text, 'lxml')
            div_pos = soup.find_all('div',{'class':'main'})
            return div_pos
        else:
            return None
    except:
        return '错误'
```

定义一个函数 craw_data()，传入 url 参数，在 main 函数中设置 url 和偏移量。则合理使用了 proxy 代理，因为猫眼电影网有反爬机制，避免同一 ip 过多地访问而锁 ip，即使如此，也很难一次性爬取这么多的数据，所以我分了两次进行爬取。

```
def main(offset):
    url = 'http://maoyan.com/board/4?offset=' + str(offset)
    html = crawl_data(url)  # 访问一次
    # print(html)
    # 下载数据
    # dum_data(html)
    # url = get_img_urls(info_url('https://www.maoyan.com/films/1200486'))
    # get_other_info(info_url('https://www.maoyan.com/films/1297'))
    show_score()
    show_actors()
    show_year()
```



从网页结构可以看出，电影内容是包含在一个’class=main’的 div 块中的，直接使用 find_all 函数找到这个 div 返回。

- **提取电影字段信息**

可以看到每个电影信息都被包含在一个 dd 内部

```
bs = BeautifulSoup(str(div_html), 'lxml')
movies = [] # 存储电影信息
for item in range(10):
    movie = {}
    movie['rank'] = bs.select('dd i.board-index')[item].string
    movie['name'] = bs.select('.name a')[item].string
    movie['score'] = bs.select('.integer')[item].string + bs.select('.fraction')[item].string
    movie['time'] = get_release_time(bs.select('.releasetime')[item].string.strip()[5:])
    movie['link'] = 'https://www.maoyan.com' + bs.select('a.image-link')[item].get('href')
    link = movie['link']
    name = movie['name']
    next_bs = info_url(link)    # 访问电影详情页面
    movie['area'] = get_other_info(next_bs)
    # 访问电影详情页面
    movie_pics = get_img_urls(next_bs) # 获得图片链接
    down_pics(name, movie_pics)   # 下载图片保存文件夹
    movie['actors'] = bs.find_all(name = 'p', attrs = {'class':'star'})[item].string.strip()[3:]
```

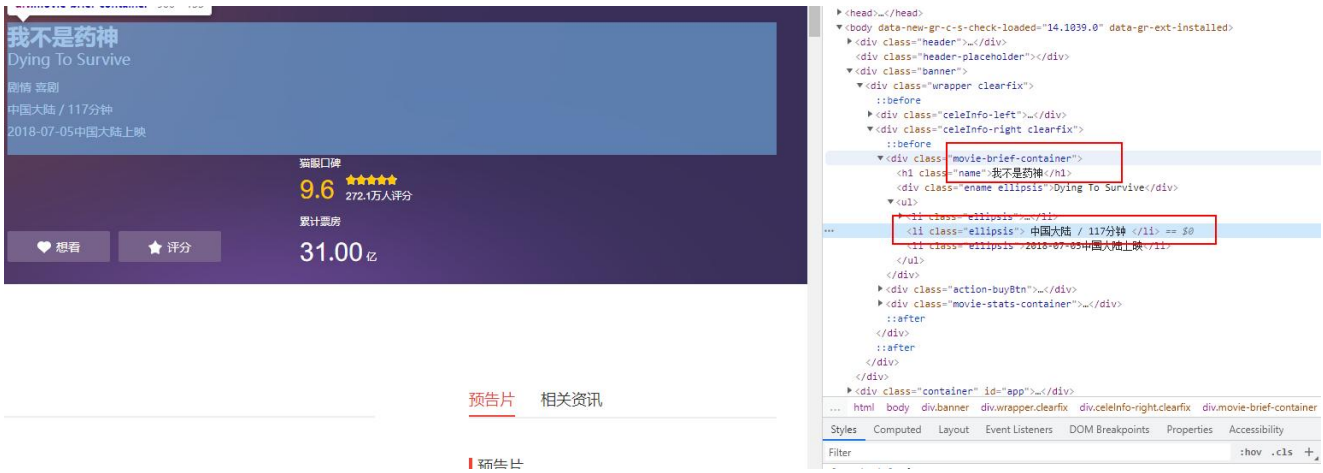
这是提取网页中电影的相关字段的代码，使用 select CSS 选择器进行选取，当然还有一些正则表达式用于提取时间和地区字段，如下，r'(.*?)(\(|$))'匹配“上映时间:2018-07-05”，re 将其分成两组，前者表示任意字符且非贪婪匹配，然后进行分片选取第五位开始的数据，即年份。

```
# 提取上映时间函数
def get_release_time(data):
    pattern = re.compile(r'(.*?)(\(|$)')
    item = re.search(pattern, data)
    if item is None:
        return '未知'
    return item.group(1)

# 提取中文字段
def get_area(data):
    item = re.search('[\u4e00-\u9fa5]*', data)
    if item is None:
        return '未知'
    # print(item)
    return item.group()
```

```
tmp_area = bs.select('.movie-brief-container ul li')[1].string.strip()
# print(tmp_area)
area = get_area(tmp_area)
```

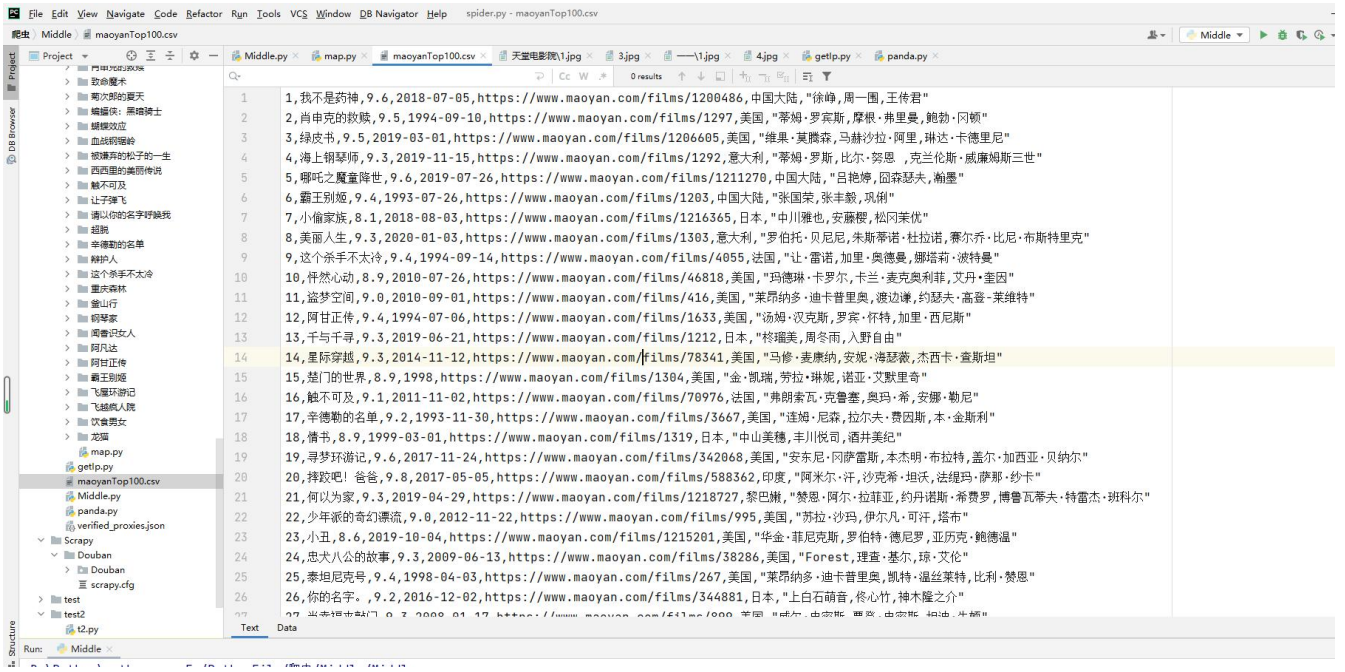
电影的上映地区需要点进电影详情链接才能爬取到，然后传到 get_area 中获取其中的中文字段即可。



- 保存数据到 csv 文件

```python
# 把字典形式的数据添加到csv文件中
def write_to_file(item):
    with open('maoyanTop100.csv', 'a', encoding='utf_8_sig', newline='') as f:
        # a为追加模式
        fieldnames = ['rank', 'name', 'score', 'time', 'link', 'area', 'actors']
        w = csv.DictWriter(f, fieldnames = fieldnames)
        w.writerow(item)
```

这里采用将字典形式的内容保存到 csv 文件中，如图

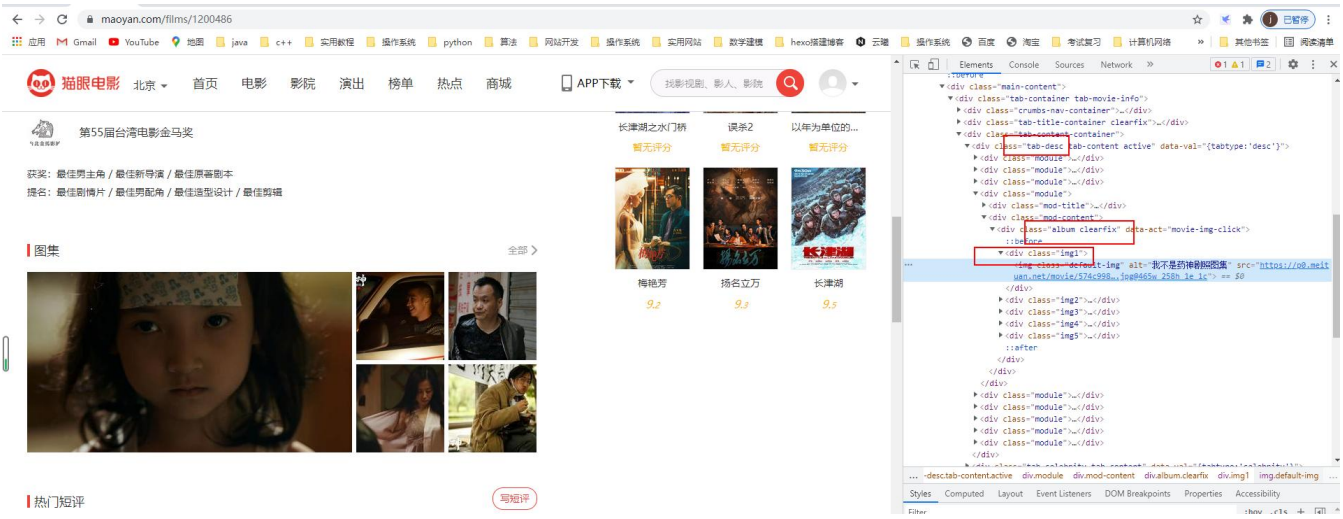


- 获取电影的图片链接

```python
#  访问电影剧照,并保存图片,先建立图片的urls集合
def get_img_urls(bs):
    # 访问网页
    # response_detail = requests.get(movies_links,headers = headers)
    # bs = BeautifulSoup(response_detail.text,'lxml')
    pic_urls = [] # 存储图片的链接集合
    main_img = bs.find(name = 'img',attrs= {'class':'avatar'})
    main_url = main_img.get('src')
    pic_urls.append(main_url)

    divs_html = bs.select('.tab-desc .album div')
    for div in divs_html:
        img_url = div.img.get('data-src')
        pic_urls.append(img_url)
    print(pic_urls) # 打印出电影的剧照集
    return pic_urls # 返回链接集合
```

首先找到电影的封面链接，获取“src”字段，并保存到图片列表中，然后在电影图集中，找到相关字段“data-src”依次保存在列表中并返回。



访问电影详情链接函数，为了避免代码冗余，直接将这多条语句封装成一个函数，返回 BeautifulSoup 对象。

```
# 访问电影详细函数、避免多次访问,返回Beautiful对象
def info_url(movies_link):
    '''
    封装请求，避免多次访问，需要人工验证
    :param movies_link:
    :return:
    '''
    response_detail = requests.get(movies_link, proxies= proxy, headers= headers)
    bs = BeautifulSoup(response_detail.text,'lxml')
    return bs
```

- 下载图片并保存到文件夹

```python
# 下载图片保存到每个电影的文件夹中
def down_pics(name,pic_urls):
    print(pic_urls)
    pic_path = '电影图片/' + name + '/'
    # 在图片->电影名->图片
    if not os.path.exists(pic_path):
        os.makedirs(pic_path) # 递归创建目录
    num = 0
    for i, pic_url in enumerate(pic_urls):
        print('----------------------') # 分割线
        try:
            pic = requests.get(pic_url,proxies=proxy, timeout=20)
            pic_str = str(i + 1) + '.jpg'
            with open(pic_path + pic_str, 'wb') as f:
                f.write(pic.content)
                print('成功下载第' + str(i+1) + '张图片' + str(pic_url))
            num = num + 1
        except Exception as e:
            print('下载' + str(pic_url) + '张图片失败')
            print(e)
            continue
```

传入一个图片链接列表，首先创建电影的图片文件夹，依次访问列表中的图片链接写入文件夹。

3、数据分析

- 电影评分分布图

```
1. # 各电影的评分饼状图
2. def show_score():
3.     matplotlib.rcParams['font.sans-serif'] = ['KaiTi']
4.     plt.style.use('ggplot')
5.     columns = ['rank', 'name', 'score', 'time', 'link', 'area', 'actors']
6.     df = pd.read_csv('maoyanTop100.csv', encoding='utf-
       8', header = None, names = columns, index_col='rank')
7.     # 8.0~8.5 8.5~9.0 9.0~9.5 >9.5
8.     size1 = 0
9.     size2 = 0
10.      size3 = 0
11.      size4 = 0
12.      plt.style.use('ggplot')
13.      plt.figure(figsize=(14,14))
14.      plt.tight_layout()
15.      for item in df.score:
16.          if 8.0 <= item < 8.5:
17.              size1 += 1
18.          elif 8.5 <= item < 9.0:
19.              size2 += 1
```

```
        elif 9.0 <= item < 9.5:
            size3 += 1
        elif 9.5 <= item:
            size4 +=1
    labels = ['8.0~8.5', '8.5~9.0', '9.0~9.5', '>=9.5']
    sizes = [size1,size2,size3,size4]
    explodes = [0,0,0.05,0]
    patches, l_text, p_text = plt.pie(sizes, labels=labels,explode = explodes, autopct='%1.1f%%', shadow=False, startangle=90, labeldistance=1)
    # 返回值得到 p_text 是饼图内部文字、l_texts 是图外的文本
    for t in l_text:
        t.set_size(40)
        t.set_rotation(30)
    l_text[1].set_rotation(90)
    for s in p_text:
        s.set_size(40)

    plt.title('top100 电影评分分布',fontsize = 45)
    plt.tight_layout()
    plt.axis('equal')
    plt.savefig('top100 电影评分分布')
    plt.show()
```

分别定义四个级别 8.0~8.5(含 8.0)、8.5~9.0(含 8.5)、9.0~9.5(含 9.0) >=9.5

- 演员电影电影数量 top10

```
# 电影数量最多的演员
def show_actors():
    matplotlib.rcParams['font.sans-serif'] = ['KaiTi']
    plt.style.use('ggplot')
    columns = ['rank', 'name', 'score', 'time', 'link', 'area', 'actors']
    df = pd.read_csv('maoyanTop100.csv', encoding='utf-8', header=None, names=columns, index_col='rank')
    actor_total = df.actors
    # print(actor_total)
    actor_list = []
    for i in actor_total.str.replace(' ','').str.split(','):
        actor_list.extend(i)
    # print(actor_list)
    # print(len(actor_list))
    # print(actor_list.count('刘德华'))
    # 去除重复的演员名
```

```
    actor = set(actor_list)
    print(actor)
    actor_all = {}
    # actor_all 字典{演员:数量}
    for i in actor: # 遍历演员 (i)
        if actor_list.count(i) > 1:
            # 选出电影数量超过 1 部的演员，并记录数量
            actor_all[i] = actor_list.count(i)
    actor_all = sorted(actor_all.items(), key = lambda actor_list:actor_list[1], reverse = True)
    # items()方法返回可遍历的(键，值)元组数组,lambda 表示按字典的第二个值排序(即电影数)
    # 元组变为字典
    actor_all = dict(actor_all[:10])
    print(actor_all)
    # 绘图
    x = list(actor_all.keys())
    y = list(actor_all.values())
    plt.bar(range(10), y, tick_label = x)
    plt.xticks(rotation = 270)
    for x,y in enumerate(y):
        plt.text(x, y + 0.1, '%s' %round(y, 1), ha = 'center', color = '#6D6D6D')
    plt.title('演员电影作品数量排名 top10', color = '#6D6D6D')
    plt.xlabel('演员')
    plt.ylabel('数量(部)')
    plt.tight_layout() # 填充整个图像区域
    plt.savefig('演员电影作品数量排名 top10.jpg')
    plt.show()
```

这个函数画的是一个柱形图。首先 pd.read_csv 读取 csv 文件的内容转化为 DataFrame 类型，因为在 csv 中同一部电影的演员不止一个，需要使用 split 函数以“，”分割开每一个演员，继续使用 set 集合进行去重，使用 sorted 函数进行排序，然后 plt.text()在每一个柱上添加注释数据。

- 电影年份与数量关系

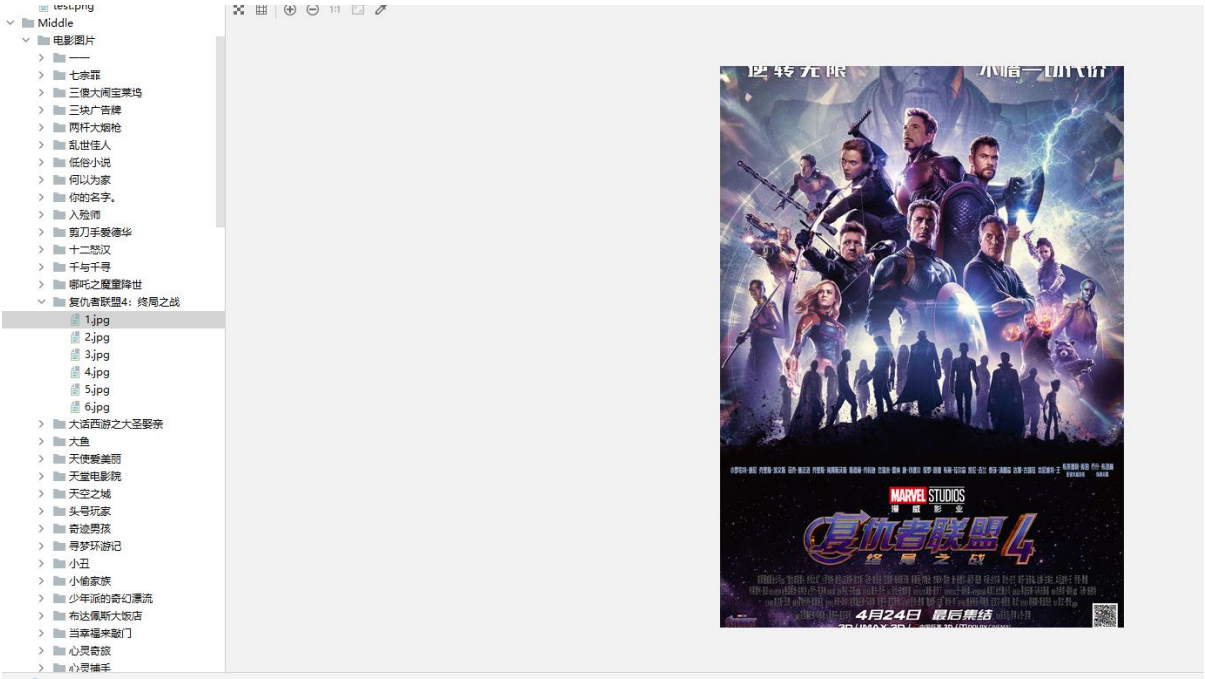
```
# 电影作品数量集中的年份
def show_year():
    plt.rcParams['font.sans-serif'] = ['SimHei']
    columns = ['rank', 'name', 'score', 'time', 'link', 'area', 'actors']
    df = pd.read_csv('maoyanTop100.csv', encoding='utf-8', header=None, names=columns, index_col='rank')
    # print(df['time'])
    df['year'] = df['time'].map(lambda x:x.split('-')[0])  # 取时期格式中的第一个数字年份作为处理结果 year 列值
    # print(df['year'])
    # 统计各年份电影数量
```

```
group_year = df.groupby('year')
group_year_mount = group_year.year.count()  # Series 类型
for x,y in enumerate(list(group_year_mount.values)):
    plt.text(x, y+0.2, '%s' %round(y,1), ha = 'center', color = "#6D6D6D", fontsize = 13)
# print(group_year_mount)
plt.title('年份电影数量分布',fontsize = 15)
plt.xticks(rotation = 270, fontsize = 10)
plt.xlabel('年份(年)')
plt.ylabel('数量(部)')
plt.plot(group_year_mount, 'o-g',ms = 5)
plt.tight_layout()
plt.savefig('年份电影数量分布.jpg')
plt.show()
```

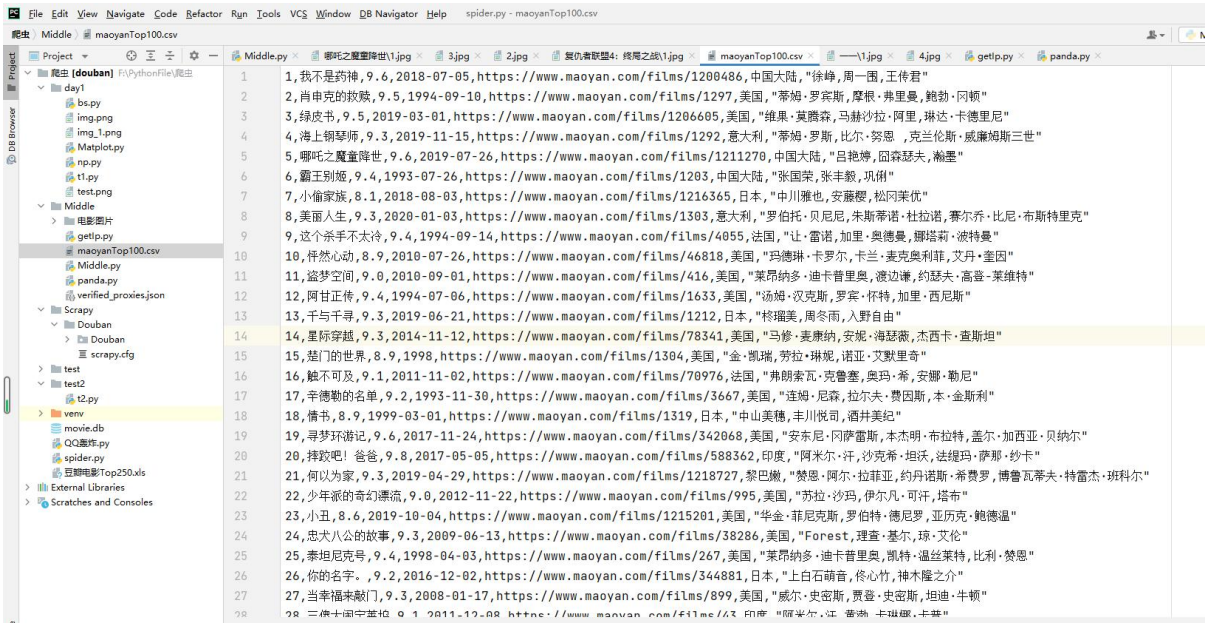
这个函数画的是折线图，这里采用了 map 函数将“time”这一列映射为“year”新的一列， df.groupby(‘year’) 对年份进行分组，每一组对应其中这一年的电影数量(year.count())，生成的 group_year_mount 是一个 Series 类型，直接传入 plt 自动生成年份—数量图像。

4、实验结果

电影图片

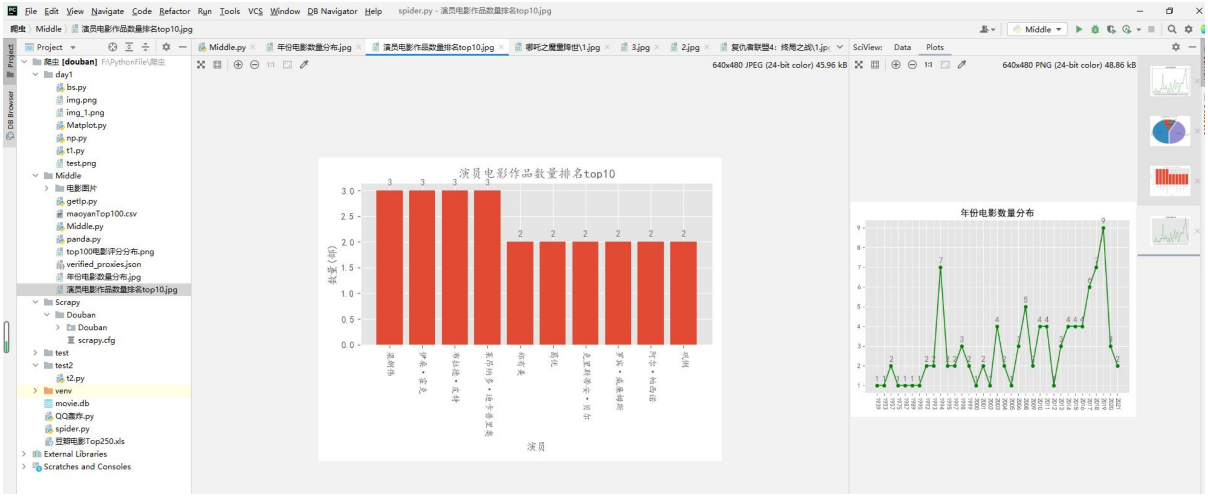


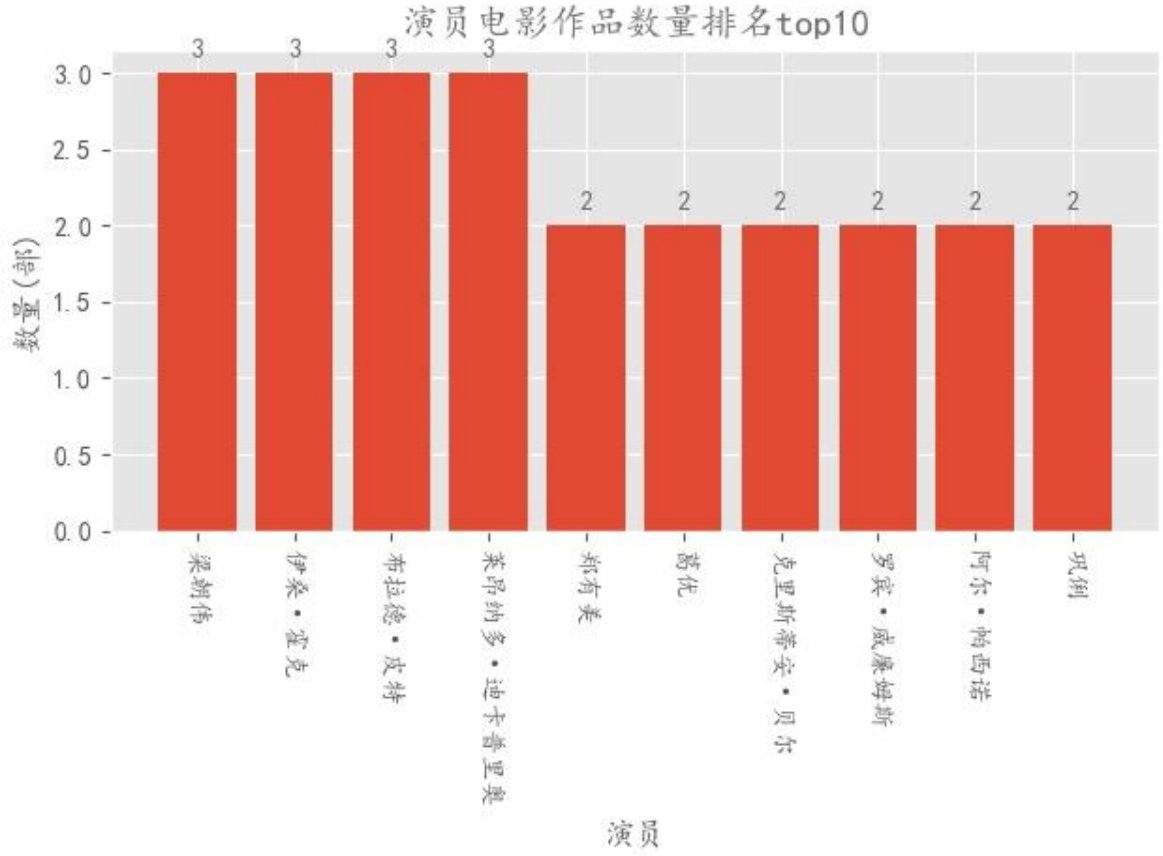
数据 csv 文件

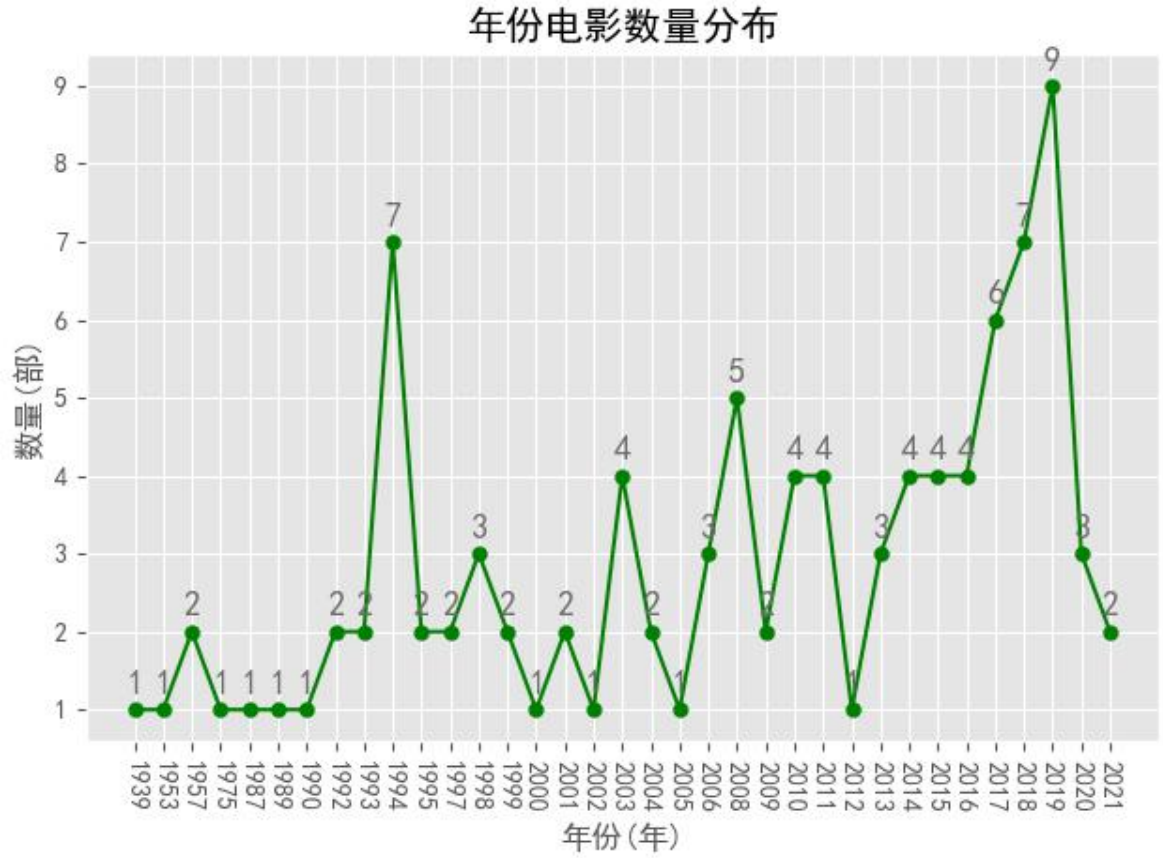


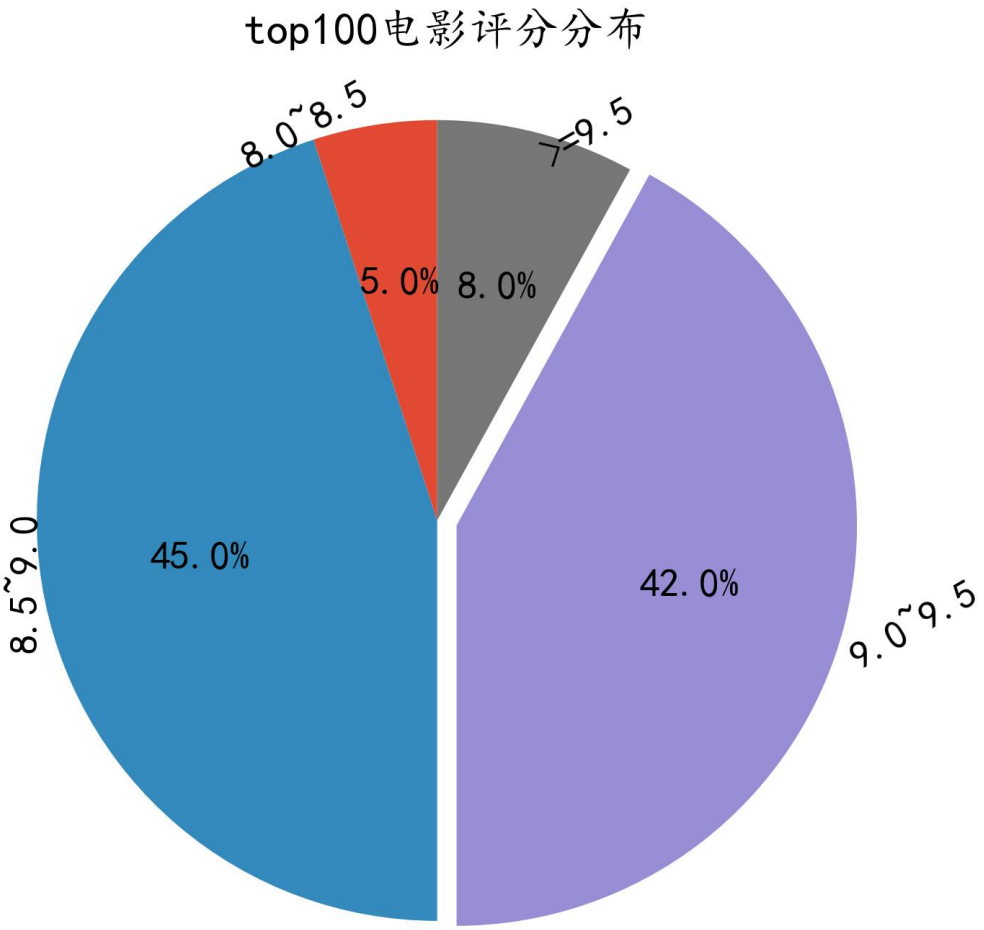
```
1,我不是药神,9.6,2018-07-05,https://www.maoyan.com/films/1200486,中国大陆,"徐峥,周一围,王传君"
2,肖申克的救赎,9.5,1994-09-10,https://www.maoyan.com/films/1297,美国,"蒂姆·罗宾斯,摩根·弗里曼,鲍勃·冈顿"
3,绿皮书,9.5,2019-03-01,https://www.maoyan.com/films/1206605,美国,"维果·莫腾森,马赫沙拉·阿里,琳达·卡德里尼"
4,海上钢琴师,9.3,2019-11-15,https://www.maoyan.com/films/1292,意大利,"蒂姆·罗斯,比尔·努恩 ,克兰伦斯·威廉姆斯三世"
5,哪吒之魔童降世,9.6,2019-07-26,https://www.maoyan.com/films/1211270,中国大陆,"吕艳婷,囧森瑟夫,瀚墨"
6,霸王别姬,9.4,1993-07-26,https://www.maoyan.com/films/1203,中国大陆,"张国荣,张丰毅,巩俐"
7,小偷家族,8.1,2018-08-03,https://www.maoyan.com/films/1216365,日本,"中川雅也,安藤樱,松冈茉优"
8,美丽人生,9.3,2020-01-03,https://www.maoyan.com/films/1303,意大利,"罗伯托·贝尼尼,朱斯蒂诺·杜拉诺,赛尔乔·比尼·布斯特里克"
9,这个杀手不太冷,9.4,1994-09-14,https://www.maoyan.com/films/4055,法国,"让·雷诺,加里·奥德曼,娜塔莉·波特曼"
10,怦然心动,8.9,2010-07-26,https://www.maoyan.com/films/46818,美国,"玛德琳·卡罗尔,卡兰·麦克奥利菲,艾丹·奎因"
11,盗梦空间,9.0,2010-09-01,https://www.maoyan.com/films/416,美国,"莱昂纳多·迪卡普里奥,渡边谦,约瑟夫·高登-莱维特"
12,阿甘正传,9.4,1994-07-06,https://www.maoyan.com/films/1633,美国,"汤姆·汉克斯,罗宾·怀特,加里·西尼斯"
13,千与千寻,9.3,2019-06-21,https://www.maoyan.com/films/1212,日本,"柊瑠美,周冬雨,入野自由"
14,星际穿越,9.3,2014-11-12,https://www.maoyan.com/films/78341,美国,"马修·麦康纳,安妮·海瑟薇,杰西卡·查斯坦"
15,楚门的世界,8.9,1998,https://www.maoyan.com/films/1304,美国,"金·凯瑞,劳拉·琳妮,诺亚·艾默里奇"
16,触不可及,9.1,2011-11-02,https://www.maoyan.com/films/70976,法国,"弗朗索瓦·克鲁塞,奥玛·希,安娜·勒尼"
17,辛德勒的名单,9.2,1993-11-30,https://www.maoyan.com/films/3667,美国,"连姆·尼森,拉尔夫·费因斯,本·金斯利"
18,情书,8.9,1999-03-01,https://www.maoyan.com/films/1319,日本,"中山美穗,丰川悦司,酒井美纪"
19,寻梦环游记,9.6,2017-11-24,https://www.maoyan.com/films/342068,美国,"安东尼·冈萨雷斯,本杰明·布拉特,盖尔·加西亚·贝纳尔"
20,摔跤吧！爸爸,9.8,2017-05-05,https://www.maoyan.com/films/588362,印度,"阿米尔·汗,沙克希·坦沃,法提玛·萨那·纱卡"
21,何以为家,9.3,2019-04-29,https://www.maoyan.com/films/1218727,黎巴嫩,"赞恩·阿尔·拉菲亚,约丹诺斯·希费罗,博鲁瓦蒂夫·特雷杰·班科尔"
22,少年派的奇幻漂流,9.0,2012-11-22,https://www.maoyan.com/films/995,美国,"苏拉·沙玛,伊尔凡·可汗,塔布"
23,小丑,8.6,2019-10-04,https://www.maoyan.com/films/1215201,美国,"华金·菲尼克斯,罗伯特·德尼罗,亚历克·鲍德温"
24,忠犬八公的故事,9.3,2009-06-13,https://www.maoyan.com/films/38286,美国,"Forest,理查·基尔,琼·艾伦"
25,泰坦尼克号,9.4,1998-04-03,https://www.maoyan.com/films/267,美国,"莱昂纳多·迪卡普里奥,凯特·温丝莱特,比利·赞恩"
26,你的名字。,9.2,2016-12-02,https://www.maoyan.com/films/344881,日本,"上白石萌音,佐心竹,神木隆之介"
27,当幸福来敲门,9.3,2008-01-17,https://www.maoyan.com/films/899,美国,"威尔·史密斯,贾登·史密斯,坦迪·牛顿"
28,三傻大闹宝莱坞,9.1,2011-12-08,https://www.maoyan.com/films/43,印度,"阿米尔·汗,黄渤,卡琳娜·卡普
```

分析图表




演员电影作品数量排名top10
数量(部)
演员
梁朝伟
伊桑·霍克
布拉德·皮特
莱昂纳多·迪卡普里奥
郑有美
葛优
克里斯蒂安·贝尔
罗宾·威廉姆斯
阿尔·帕西诺
巩俐


年份电影数量分布
数量(部)
年份(年)



六、参考资料

[1] 朝乐门 著.Python 编程从数据分析到数据科学.电子工业出版社.2019.1

[2] [美] 阿曼多·凡丹戈（Armando Fandango） 著，韩波 译. Python 数据分析(第 2 版). 人民邮电出版社.2018.6

[3] 嵩天，礼欣，黄天羽 著.Python 语言程序设计基础（第 2 版）. 高等教育出版社.2017.2