Python 在疫情统计数据可视化中的应用研究*

滨州医学院卫生管理学院 曹振丽 王立业 赵晓娜 曹高芳△

【中图分类号】 R181.2 【文献标识码】 A DOI 10.11783/j.issn.1002-3674.2024.03.027

在流行病学研究中,人们常常使用统计地图来描述某种疾病在某个地区、某一时间的分布即统计数据的可视化问题。统计学者使用统计分析商用软件 SAS 和 R 语言较多,但是 SAS 软件默认设置的绘图精度不高,调整需要用户翻阅帮助手册进行深入学习;而 R 语言绘图适用于那些熟悉 ggplot2 制图的人员,新手没有 ggplot2 制图基础,学习该软件比较困难[1]。上述的操作都比较复杂,对于新人而言不容易上手。

Python 语言简单易学,语法简洁,功能强大,代码易于阅读和理解,新手能够快速上手编写代码,被广泛用于 Web 应用程序开发、科学计算和数据分析与处理、可视化、图像处理等众多领域,为学科交叉融合提供技术支持。Python 语言通过其各种内置的标准库与可下载安装的第三方库,可以实现数据的爬取、解析、保存、可视化等操作,譬如 Python 调用地图定位软件 API 在地图上添加相关标记;Numpy 的计算包实现对数据矩阵的存储和处理、实现对数据的可视化和预处理[2]。

本研究前期完成了利用 Python 爬虫爬取卫健委网站相应时间段的数据,并分析网页数据文本形式,编辑对应的正则表达式,实现对疫情数据的清洗和存储。本论文中将以 2019 年 12 月至 2023 年 1 月互联网官网公布的新冠疫情的数据为例,利用 Python 语言进行编程,调用相关的类库,在前台绘制出各种样式的图形,直观形象的实现疫情数据的可视化。

程序介绍

1.软件安装

根据电脑位数登录 Python 官网 https://www.python.org 下载相应安装包,解压到想要安装的目录,安装时务必勾选 Add Python 3.11 to PATH,安装完成后点击“关闭路径限制”按钮,即可完成安装。使用命令提示符验证 Python 是否安装成功,在命令提示符框中输入 Python 后回车,显示 Python 3.11,表示安装成功。本文代码运行环境为 Windows 10 操作系统,需要安装的第三方库 echarts、wordcloud、pyecharts 使用 pip 工具进行安装。

2.疫情数据可视化处理

利用 echarts 与 pyecharts 来实现数据的可视化,二者提供了常规的折线图、柱状图、散点图等,还提供了用于统计的盒形图及用于地理数据可视化的地图等[3]。在使用之前先在 html 文件中导入 echarts.js 文件和 jquery.js 文件,为要创作的图表构建一个合适的容器,在 Apache ECharts 官网的实例中挑选能够给用户带来舒适、清晰体验的数据图表。本论文的可视化部分以全国各地疫情数据统计、山东省疫情状况统计、山东省当日新增与治愈人数对比折线图、山东省各市累计确诊人数柱状图、山东省各城市 2022 年 3 月至 12 月治愈人数的数据玫瑰图等来进行编码展示,其中不同的数据灵活采用图表形式。

(1)全国各地疫情数据统计

将数据库 pro_add_1 表中数据进行累加,累加方式为将 add_num 列按照 pro_name(省名称)分组进行累加,这样就得到 2022 年 12 月 21 日到 2023 年 1 月 8 日各省累计确诊人数,再加上各省之前公布的累计确诊人数就可以得到各省总体累计确诊人数。通过 pyecharts 将各省累计确诊人数呈现出来。利用分段式可视化组件用不同颜色表示各省确诊人数范围,确诊人数的多少决定颜色的深浅。鼠标悬停在某一省份会显示该省份的名称和累计确诊人数[4],可以通过滑动鼠标滚轮对地图进行放大缩小。

实现全国各地疫情数据在网页中展示的核心部分代码如下:

```
map = Map()
map.add("各省份确诊人数", data, "china")
map.set_global_opts(
title_opts=TitleOpts(title=f"全国疫情地图"),
visualmap_opts=VisualMapOpts(
        is_show=True,  # 是否显示
        is_piecewise=True,  # 是否分段
        pieces=[
```

*基金项目:国家社科基金项目(18BGL244);教育部产学合作协同育人项目(220905259190632);山东省教育科学研究课题(2021JXY031);烟台社科规划研究项目(YTSK-2022-054);滨州医学院校级课题(20SKNY02)

△通信作者:曹高芳,E-mail:caogaofang2003@163.com

```
            {"min": 1, "max": 99, "lable": "100~250 人", "color": "#CCFFFF"},
            {"min": 100, "max": 250, "lable": "100~250 人", "color": "#FFF8DC"},
            {"min": 251, "max": 500, "lable": "251~500 人", "color": "#FFFF99"},
            {"min": 501, "max": 999, "lable": "501~999 人", "color": "#FFDEAD"},
            {"min": 1000, "max": 4999, "lable": "1000~4999 人", "color": "#FF9966"},
            {"min": 5000, "max": 9999, "lable": "5000~99999 人", "color": "#FF6666"},
            {"min": 10000, "max": 99999, "lable": "10000~99999 人", "color": "#CC3333"},
            {"min": 100000, "lable": "100000+", "color": "#990033"},
                ]      ))
    map.render(f"全国疫情数据展示.html")
                ]      ))
```

(2)山东省疫情状况统计

将数据库 shandong 表中数据进行整合,整合方式为:将 day_add_comf、day_cure、die_num 中数据进行累加,得出山东省在 2022 年 3 月 12 日到 2022 年 12 月 16 日累计确诊人数、累计治愈人数、累计死亡人数

将数据库中 city_in_shandong 表中数据进行整合,整合方式为:将 comf_num 列按照 city_name 列(城市名称)分组进行累加,由此得出山东省各市在 2022 年 3 月 12 日到 2022 年 12 月 16 日累计确诊人数。通过 pyecharts 将各市累计确诊人数,以山东省地图的形式呈现出来。利用分段式可视化组件用不同颜色表示各市确诊人数范围,确诊人数的多少决定颜色的深浅。鼠标悬停在某一城市会显示该城市的名称和累计确诊人数,可以通过滑动鼠标滚轮对地图进行放大缩小。

核心部分代码如下:

```
    map = Map()
    map.add("城市累计确诊人数", pro_list, f"{pro}")
    map.set_global_opts(
    title_opts=TitleOpts(title=f"{pro}疫情地图"),
    visualmap_opts=VisualMapOpts(
    is_show=True,
    is_piecewise=True,
    pieces=[
      {"min": 1, "max": 99, "lable": "1~99 人", "color": "#CCFFFF"},
    {"min": 100, "max": 250, "lable": "100~250 人","color": "#FFF8DC"},
    {"min": 251, "max": 500, "lable": "251~500 人", "color": "#FFFF99"},
    ]))
```

(3)山东省当日新增与当日现存人数对比折线图

通过折线图能够直观的看出数据的增长或下降,本论文的折线图的横坐标都是由日期组成,而纵坐标由每日新增及新增确诊人数、新增治愈人数、新增死亡人数等数据组成。可视化部分的内容并不是一成不变的,用户并不只是通过视觉感受可视化图表的效果,用户对可视化界面的反馈以及对界面的操作也属于可视化的一部分,使用 Echarts 提供的交互组件来提高用户的体验感,系统中的折线图使用 markPoint 和 markLine 组件来增加折线图的功能[5]。

以全国 2022 年 12 月 21 日至 2023 年 1 月 8 日当日新增与当日现存人数对比折线图为例,连接数据库后,将 nation_data 表中 date 列按照升序进行排列,选择 date、add_num、day_cure 进行输出,核心部分代码为:

```
    cursor_1 = conn.cursor()
    conn.select_db("yq")      # 选择数据库
    sql = '''select date,now_comf,add_num,day_cure from nation_data order by date asc '''
    cursor_1.execute(sql)      # 执行 SQL 语句
    data = cursor_1.fetchall()
    for item in data:          # 将数据库中的数据通过遍历循环保存到列表
        date.append(str(item[0]))
        now_comf.append(item[1])
        add_comf.append(item[2])
        day_cure.append(item[3])
    cursor_1.close()
    conn.close()
```

得到数据后,进入制表环节,折线图中当日新增人数为蓝色线条,当日治愈人数为绿色线条,鼠标悬停时显示日期、该日新增人数、该日确诊人数,虚线表示为两种数据的平均值,部分实现代码为:

```
    series:[   {      name:'当日新增确诊人数',
        type:'line',
        data:{{add_comf|tojson}},
        markPoint:{       data:[
            { type:'max', name:'Max'},
            { type:'min', name:'Min'}          ]
        }, # 设置 markPoint,分别标记为每条折线的最大值与最小值
        markLine:{
          data:[{ type:'average', name:'Avg'}]
        }   }   ], # 设置 markLine,分别标记两条折
```

线的平均值

山东省当日新增与当日现存人数对比折线图在页面中的展现形式如图 1 所示。

图 1 山东省当日新增与当日现存人数对比折线图

(4)山东省各市累计确诊人数柱状图

柱状图可以直观的看出某一对象对应的数量,方便各个对象的比较,本系统中的柱状图用于比较各省或各市的确诊人数、治愈人数、死亡人数等。柱状图调用数据库的方法与折线图相似,只是读取不同的数据库表,配置不同的图表类型、格式和数据等。以山东省各市累计确诊人数柱状图为例,设计形式为蓝色柱状条,鼠标划过时会出现阴影以及显示城市名称、累计确诊人数,由图 2 山东省各市累计确诊人数柱状图可以看出济南为最高,说明济南疫情状况最为严重,青岛紧随其后,其中潍坊与东营数量垫底,说明其并未经历较为严重的疫情。

柱状图核心部分实现代码如下:

```
series:[    {
        name:'累计确诊人数',
        type:'bar',
        barWidth:'60%',
        data:{{comf_num|tojson}}
    }]
```

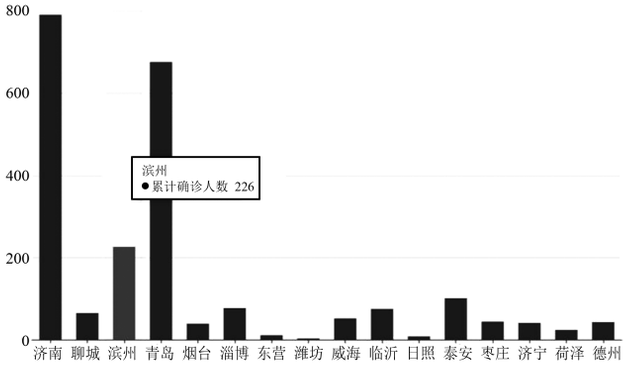

图 2 山东省各市累计确诊人数柱状图

(5)山东省各城市 2022 年 3 月至 12 月治愈人数的数据玫瑰图

玫瑰图一般使用圆弧的半径来表示数据大小,强调数据大小的对比。其实现方法与柱状图相似,只是选取不同的数据库表、不同的图表类型、格式等。其在网页中的展现形式如图 3 所示。

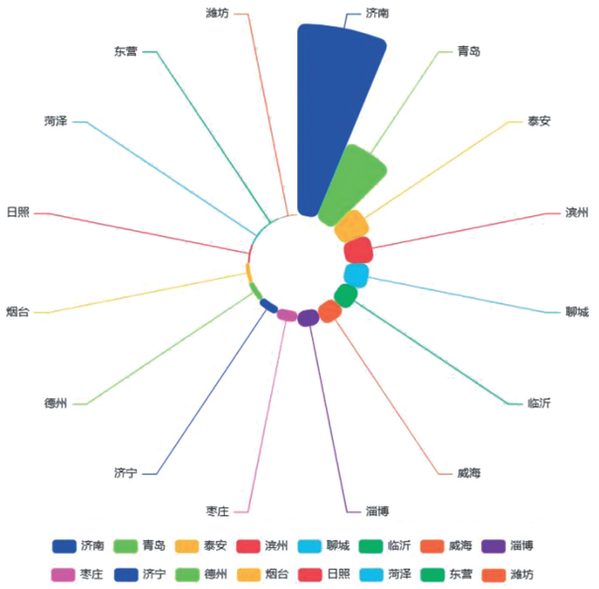

图 3 山东省各城市 2022 年 3 月至 12 月治愈人数的数据玫瑰图

玫瑰图中半径越大表示数量越多,从图 5 的玫瑰图中可以看出济南的半径最大,意味着治愈人数最多;其次是青岛、泰安、滨州、聊城、临沂、威海、淄博、枣庄、济宁、德州、烟台、日照、菏泽、东营、潍坊。其数据与累计确诊柱状图相照应,城市累计确诊人数越多其累计治愈人数也就越多。

(6)疫情热点词词云图

由疫情以来的各种热点词组成疫情热点词词云图,词语出现的频率越高,显示的字体就越大、越突出,意味着该关键词非常的重要,词云图可以让浏览者快速抓住重点。在该部分主要运用 jieba、matplotlib、PIL、numpy、wordcloud 库中的一些方法,其中 jieba 库用于将收集到的各种句子分词,运用 matplotlib 库中的 pyplot 进行绘图,将数据可视化,PIL 中的 Image 用于图片处理,numpy 用于矩阵运算,最重要的是 wordcloud 中的 WordCloud 用于制作词云。

核心实现代码如下:

```
# 分词
cut = jieba.cut(text)
str = ''.join(cut)
print(len(str))
img = Image.open(r'D:\python学习\pythonProject5\spider\tree.jpg')
img_array = np.array(img)   # 将图片转换为数组
wc = WordCloud(
        background_color='white',
        mask=img_array,
```

```
        font_path = " msyh.ttc"          # 选择字体
    )
    wc.generate_from_text(str)
    # 绘制图片
    fig = plt.figure(1)
    plt.imshow(wc)
    plt.axis('off')                  # 是否显示坐标轴
    # 输出词云图片到文件,设置清晰度
    plt.savefig(r'D:\python 学习\flask_dome\Static\assets\images\word.jpg',dpi=500)
```

词云图在网页中的显现形式如图 4 所示。

图 4　疫情热点词词云图

结　果

由图 1 到图 4 的各种形式的疫情统计数据可视化图可知,利用 Python 语言,编程调用 Echarts 与 PyEcharts 库,可以简捷高效地绘制各种动态形象的疫情数据曲线,直观展示流行病学分布情况等,图像清晰,数据准确,结果可重复,具有专业性,是统计学和流行病学工作者办公的的利器。

结　语

随着大数据时代的到来,在以流行病学与卫生统计学等疾控领域,对疾病暴发、监测方面的数据分析与可视化需求将日益增多,但缺乏简便快速揭示特征,发现危险因素的分析工具,在实际工作中疾病暴发调查数据的快速分析仍是一个薄弱环节[6]。大数据具有巨大的价值,深挖数据中蕴含着的奥秘,透过现象看本质,可以描述事件的发展历程。而将大数据信息可视化,以图形和色彩的形式将关键数据和特征值,直观形象的呈现给用户,病案统计人员的工作效率和能力都会有较大的提升[6],有利于全方位多角度的对数据进行解析。

本论文主要研究利用 Python 语言编程,灵活高效的把疫情统计数据以生动具体的图表形式将其展现出来,为用户提供了一个浏览疫情信息的平台,为医务工作者提供可研究数据,为用户的学习和生活提供便利,具有实用性。建议相关部门加强对流行病学与卫生统计学等科研工作者的 Python 语言应用培训,进一步提高从业人员对疾病数据分析处理与绘制的能力。

参　考　文　献

[1] 邓晓伟,陈云峰,刘红伟,等.R 与 SAS 软件在绘制统计地图方面的应用研究.中国卫生统计,2019,36(5):798-800.
[2] 陈优良,龚曲.新工科背景下多学科融合 Python 程序设计探索.计算机教育,2022(8):118-122.
[3] 彭顺生.基于 ECharts 的肺炎疫情数据处理与可视化.计算机时代,2020(7):47-49,53.
[4] 崔蓬.ECharts 在数据可视化中的应用.软件工程,2019,22(6):42-46.
[5] 孙亚军,肖伦.R 软件在分析疾病暴发调查数据中的应用.中国卫生统计,2020,37(5):792-794,799.
[6] 马赫,葛丽,刘颖.复杂检索需求下 SQL 在医院统计中的应用思路.中国卫生统计,2020,37(1):136-140.

(责任编辑:郭海强)

(上接第 454 页)

[17] Zhi Y, Lin Z, Ma J, et al. Distinguish the Role of Radiotherapy From Chemoradiotherapy for Gastric Cancer With Behavior of Metastasis-Indolent in Lymph Node. Technol Cancer Res Treat, 2020, 19(2): 1533-1541.
[18] Shi J, Kang W, Tang Y, et al. Adjuvant Chemoradiotherapy Versus Adjuvant Chemotherapy for Stage III Gastric or Gastroesophageal Junction Cancer After D2/R0 Resection. Front Oncol, 2022, 12(7): 916937-916944.
[19] He Z, Xu JG. Evaluation of the efficiency and safety of combined chemotherapy and molecular-targeted therapy in the treatment of advanced gastric cancer: A protocol for systematic review and meta-analysis. Medicine (Baltimore), 2021, 100(45): e27557.

(责任编辑:郭海强)