基于约翰霍奇金斯大学数据集和腾讯公开数据的新型冠状病毒肺炎感染情况分析

——https://github.com/CSSEGISandData/COVID-19——

——https://api.inews.qq.com/newsqa/v1/automation/foreign/country/ranklist——

姓名：雷浩波 班级：2019211121 学号：2019210649

上交时间：

摘要：

新型冠状病毒肆虐全球，全人类的生命健康受到威胁，社会无法维持其正常秩序，分析疫情走势，制定防疫计划的工作就显得尤为重要。该项目发挥了python这门语言在数据科学方面的极大优势，调用相关的第三方库，用简洁的代码实现了数据分析和数据可视化。项目基于约翰霍奇金斯大学的新型冠状病毒数据集进行了一系列的数据分析，生成了大量方便阅读的各样式图表，展现了疫情的时间走势和空间爆发点，为相关传染病学研究打下基础。项目调用了腾讯公开的疫情数据，绘制了动态地图，也方便了群众对于疫情的总体情况有一个认识和把握。

正文:

一、项目框架与设计

1.1 目的

本项目通过对约翰霍奇金斯大学新型冠状病毒数据集的数据分析和数据可视化，得出不同时间不同国家的感染情况，死亡情况，治愈情况，对疫情的整体走势进行统计，直观得展现各个地区的疫情状况。通过对腾讯公开疫情数据的采集处理与分析，绘制全球感染人数分布地图，用不同颜色表示疫情状况。在这些工作的基础上，可以方便医学研究者进行传染病学分析，得出病毒的危险性，传染能力，建立流行病学模型，预测疫情走势。数据可视化部分也可以作为疫情防控工作者的参考，不断调整防控的重点地区，因地制宜的制定疫情防控方法。将疫情统计结果以简单易懂的方式呈现出来，保证公民对国家公共卫生安全状况的知情权。

1.2整体实现的描述

此项目完成了以下的工作（与项目文件名称相对应）：

世界总感染、死亡、痊愈情况排名分析

世界总感染、死亡、痊愈情况散点图分析

世界总感染、死亡、痊愈情况折线图分析

世界每日感染、死亡、痊愈情况柱状图分析、表格分析

世界受感染地区折线图

世界疫情地图可视化

二、项目实现

2.1技术应用与环境

该项目在基于约翰霍奇金斯数据集的部分主要应用了pandas、numpy、matplotlib、os等python常用第三方库。而在基于腾讯公开疫情数据的部分（世界疫情情况地图可视化）主要应用了json、requests、jsonpath、pyecharts等库。

2.2实现方式

该项目在基于约翰霍奇金斯数据集的部分使用循环体引入了数据集中一月二十二日到三月十八日的疫情情况统计CSV文件。通过sort\_values方法生成排名，并通过display\_html方法输出了世界总感染、死亡、痊愈情况的排名分析（前五名）。在上述引入文件的同时也将文件名进行切片处理并借助datetime生成了一个日期列表，作为后期绘图的横轴数据。借助sum方法统计出了每一天世界总的感染、死亡、痊愈人数，生成了对应的三个列表作为纵轴数据。在横纵轴数据的基础上，调用相应的绘图函数，分别完成了世界总感染、死亡、痊愈情况的散点图、折线图绘制。除此以外项目还使用groupby方法对每日各地区的感染情况做了统计，绘制了相应的表格和柱状图。

该项目在基于腾讯公开疫情数据的部分通过模拟浏览器访问来获取数据，对数据格式进行了相应的处理，为原先数据中的英文地名加上了中文名称，最后借助于pycharts库实现了动态地图的效果。

2.3核心代码

基于约翰霍奇金斯大学数据集部分（以其中一个文件为例）：

1. #引入所需要的的库
2. **import** pandas as pd
3. **import** numpy as np
4. **import** matplotlib.pyplot as plt
5. **import** os
6. **from** datetime **import** datetime

9. #引入文件并打印引入的文件名
10. root='C:\\Users\\thinkpad\\Desktop\\CSSE COVID-19'
11. file\_names=os.listdir(root)
12. file\_ob\_list=[]
13. Datelist=[]
14. **for** file\_name **in** file\_names:
15. fileob=root+'\\'+file\_name
16. file\_ob\_list.append(fileob)
17. Datelist.append(file\_name[0:10])
18. **print**(Datelist)

21. #绘制各国家地区每日感染情况散点图；出每日感染，死亡，康复数据；绘制每日国家地区情况表格
22. Confirmlist=[]
23. Recoverlist=[]
24. Deathlist=[]
25. i=0
26. **print**("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")
27. **for** file\_ob **in** file\_ob\_list:
28. df1=pd.read\_csv(str(file\_ob),keep\_default\_na=0,dtype=str,index\_col='Country/Region')
29. df1['Confirmed']=df1['Confirmed'].astype(float)
30. sum\_Confirmed= df1['Confirmed'].sum()
31. **print**("sum\_Confirmed is",sum\_Confirmed)
32. df1['Deaths']=df1['Deaths'].astype(float)
33. sum\_Deaths=df1['Deaths'].sum()
34. **print**("sum\_Deaths is",sum\_Deaths)
35. df1['Recovered']=df1['Recovered'].astype(float)
36. sum\_Recovered=df1['Recovered'].sum()
37. **print**("sum\_Recovered is",sum\_Recovered)
39. sum\_province=df1.groupby('Country/Region').sum()
40. **print**(sum\_province)
41. x=df1.index.values
42. y=df1["Confirmed"].values
43. pl=plt.bar(x,y,width=0.8,log=True)
44. plt.xticks(size='small',rotation=90,fontsize=5)
45. plt.show()
46. i+=1

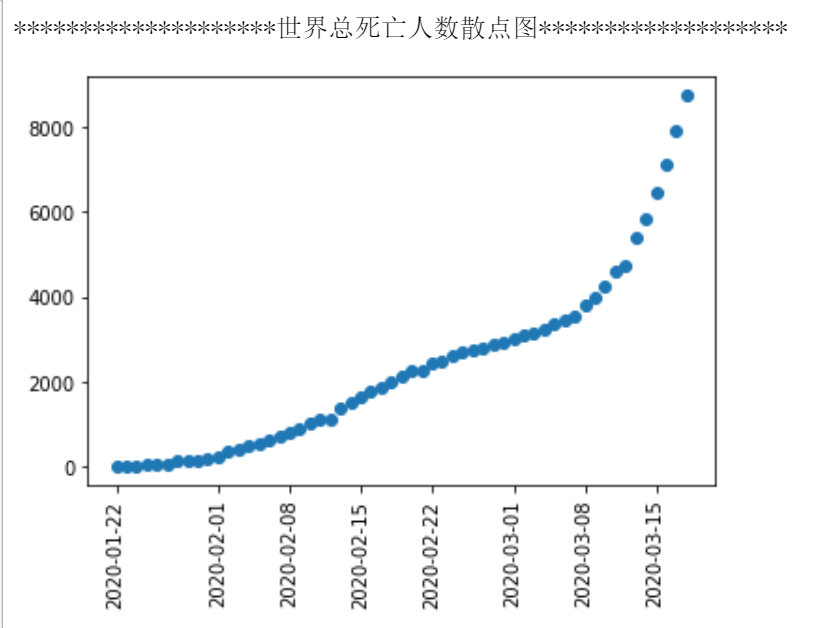
基于腾讯公开数据的部分：

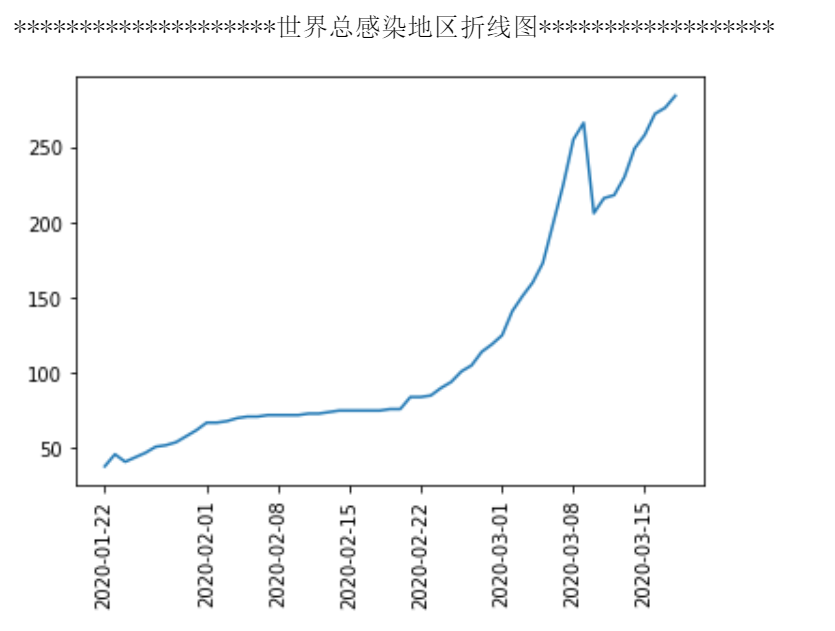
1. # 目标网址
2. url = 'https://api.inews.qq.com/newsqa/v1/automation/foreign/country/ranklist'
3. # 模拟浏览器实现访问
4. resp = requests.post(url).text
5. data = json.loads(resp)
6. name = jsonpath.jsonpath(data, "$..name")
7. **print**(len(name))
8. confirm = jsonpath.jsonpath(data, "$..confirm")
9. **print**(len(confirm))
10. a = zip(name, confirm)
11. map\_ = Map(opts.InitOpts(width='1200px', height='600px')).add(series\_name="世界各国确诊病例",  # 设置提示框标签
12. data\_pair=a,  # 输入数据
13. maptype="world",  # 设置地图类型为世界地图
14. name\_map=nameMap,  # 添加映射
15. is\_map\_symbol\_show=False  # 不显示标记点
16. )
17. # 设置系列配置项
18. map\_.set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=False))  # 不显示国家名称
19. # 设置全局配置项
20. #map\_.set\_global\_opts(title\_opts=opts.TitleOpts(title="国外疫情情况"),  # 设置图标题
21. #自动分组无法全面覆盖数据，所以后期采用了手动分组的方法
22. #visualmap\_opts=opts.VisualMapOpts(max\_=3000000, is\_piecewise=True))  # 显示图例
23. map\_.set\_global\_opts(title\_opts=opts.TitleOpts(title="国外疫情情况"),  # 设置图标题
24. visualmap\_opts=opts.VisualMapOpts(pieces=[  # 自定义分组的分店和颜色
25. {"min": 900000, "color": "#800000"},
26. {"min":500000,"max":900000,"color":"ff0033"},
27. {"min": 50000, "max": 500000, "lable":'0.15~0.19', "color": "#AA0000"},
28. {"min": 10000, "max": 50000, "color": "#CC0000"},
29. {"min": 1000, "max": 10000, "color": "#FF0000"},
30. {"min": 0, "max": 1000, "color": "#FF3333"},
31. {"max": 0, "color": "#FFCCCC"},
32. ],
33. is\_piecewise=True))  # 显示分段式图例
34. map\_.render("世界疫情情况.html")

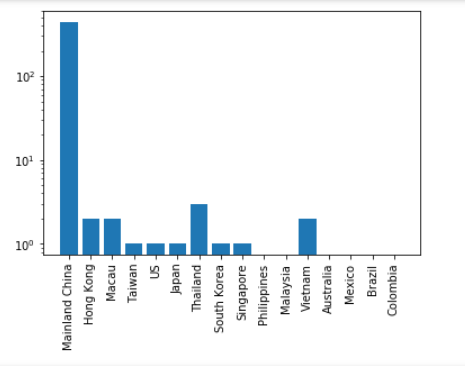
三、结果

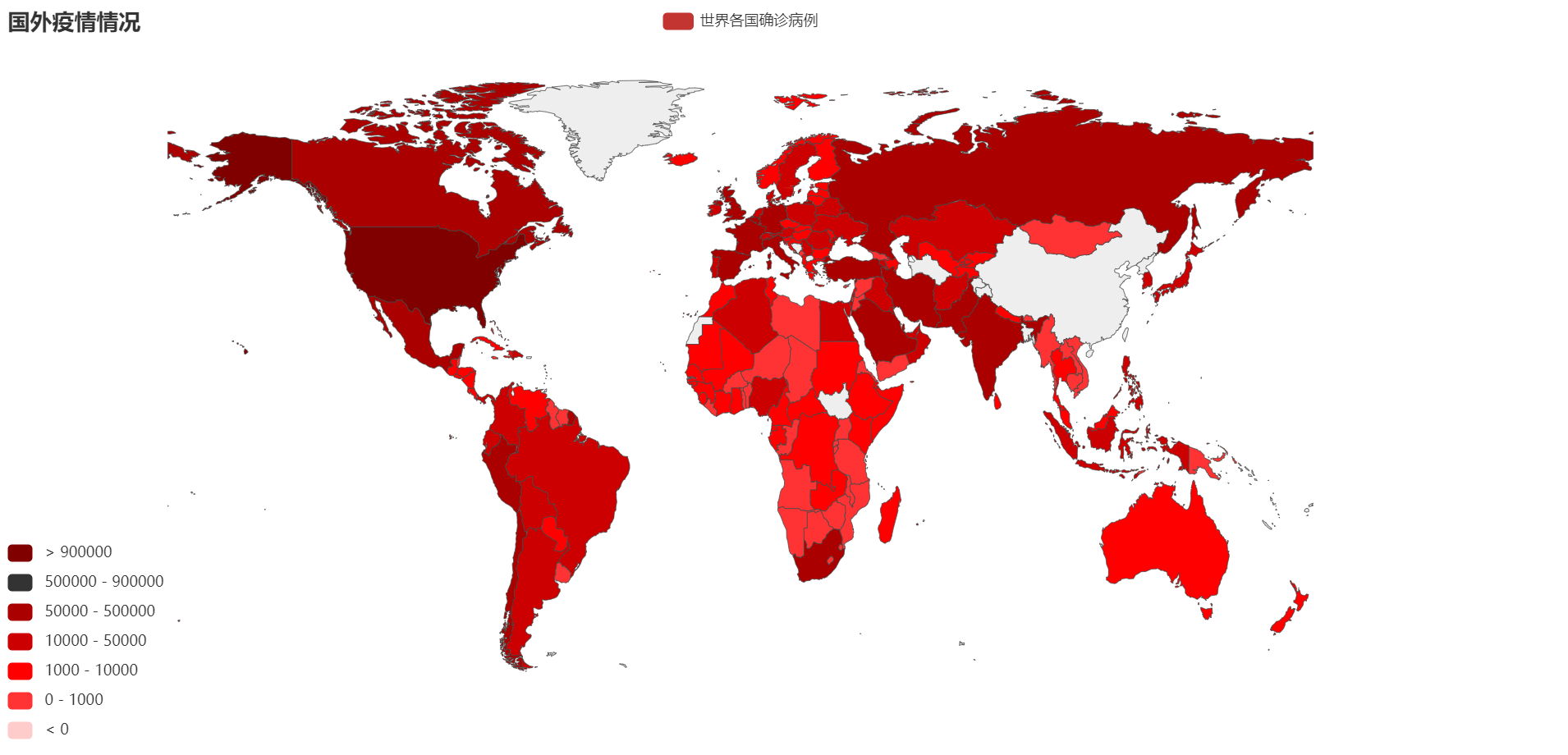
由于该项目生成了大量图表，在报告中仅展示部分具有代表性的，完整数据保存在项目文件中，并附有生成过程。注意，动态地图不便在word中展现HTML文件，项目根目录下附有相应文件。











四、未来的展望和讨论

当下国内疫情已经得到了很好的控制，可以说我国已经在这场抗疫攻坚战中取得了胜利，但国外的疫情状况仍不容乐观，美国的感染人数已经达到了两百万人左右。希望此项目能够为疫情分析与疫情防控工作作出贡献，使人类更好的了解疫情的传播规律，通过制定合理的抗疫方法，阻止疫情进一步蔓延，也为以后的公共卫生安全保障积累宝贵经验。

在此项目的基础上，可以进一步丰富数据范围，用更大的样本建立更加精确的模型，让模型的结果更加贴合实际。还可以对项目内部的操作进行进一步的封装，使该项目能够用于分析人类所遇到的各种传染病，形成传染病的数据分析与数据可视化体系。

参考文献

[1]<https://docs.python.org/zh-cn/3/> python官方文档

[2]<https://pandas.pydata.org/> pandas开发文档

[3]<https://matplotlib.org/> matplotlib开发文档

[4]<http://pyecharts.herokuapp.com/> pyecharts开发文档