

使用Python绘制疫情地图

一、应用场景

对于业务地域比较广的公司，比如地厂商、连锁店等，可以对该代码适当修改之后，对各地区收集到的数据进行地图绘制，通过不同的颜色展示，更加直观的展示各个地区的业务情况。

二、项目代码与结果展示

项目代码

```
1 # 爬取数据
2 import requests, json
3 url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5' # 数据接口
4 data = json.loads(requests.get(url).json()['data']) # 读取数据转换为 JSON 格式
5 update_time = data['lastUpdateTime'] # 更新时间
6 China = data['areaTree'][0]['children'] # 全国各地具体数据
7
8 map_data = []
9 for i in China :
10     province = i["name"] # 获取省份名称
11     province_confirm = i["total"]["confirm"]
12     map_data.append((i["name"],province_confirm)) # 保存省份名称和该省确诊人数
13
14 # 绘制地图
15 from pyecharts.charts import Map
16 from pyecharts import options as opts
17 data_label = "确诊"
18 map_title = "全国疫情图"
19 c = (Map() # 声明一个map对象
20     .add(data_label, map_data, "china") # 添加数据
21     # 设置标题和颜色
22     .set_global_opts(title_opts=opts.TitleOpts(title=map_title),
23                     visualmap_opts=opts
24                         .VisualMapOpts(split_number=6, # 分成6个区间
25                                         is_pieewise=True, # 将颜色图例分开展示
26                                         pieces=[{"min":1,"max":9,"label":"1-9人","color":"#ffefd7"}],
```

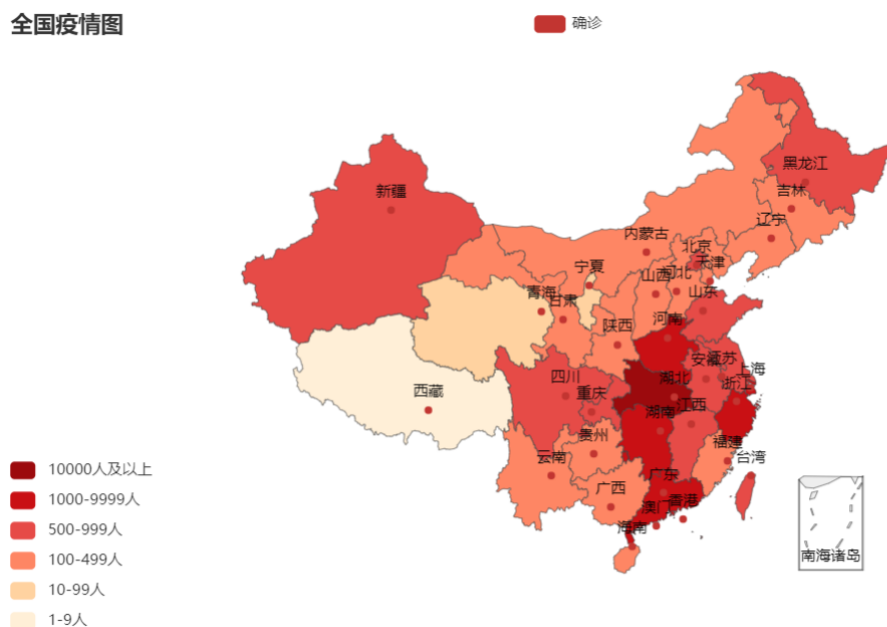
```

27         {"min":10,"max":99,"label":"10-99人","color":"#ffd2a0"},
28         {"min":100,"max":499,"label":"100-499人","color":"#fe8664"},
29         {"min":500,"max":999,"label":"500-999人","color":"#e64b47"},
30         {"min":1000,"max":9999,"label":"1000-9999人","color":"#c91014"},
31         {"min":10000,"label":"10000人及以上","color":"#9c0a0d"}
32     ]))
33 )
34
35 c.render("全国实时疫情.html") # 生成html文件

```

2.2 结果展示

全国疫情图



三、项目操作及解析

注意：pyecharts库是一个第三方库，需要先安装库和相关的地图。（Mac系统改用pip3）

```

pip install pyecharts      # 安装地图文件
pip install echarts-countries-pypkg    # 全球国家地图
pip install echarts-china-provinces-pypkg # 中国省级地图
pip install echarts-china-cities-pypkg  # 中国市级地图

```

3.1 爬取数据

该部分的代码仅适用于爬取疫情数据。

```
1 # 爬取数据
2 import requests, json
3 url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5' # 数据接口
4 data = json.loads(requests.get(url).json()['data']) # 读取数据转换为 JSON 格式
5 update_time = data['lastUpdateTime'] # 更新时间
6 China = data['areaTree'][0]['children'] # 全国各地具体数据
7
8 map_data = []
9 for i in China :
10     province = i["name"] # 获取省份名称
11     province_confirm = i["total"]["confirm"]
12     map_data.append((i["name"], province_confirm)) # 保存省份名称和该省确诊人数
```

代码解析：从数据接口爬取疫情的json格式数据，然后从逐一取出城市和确诊人数。

3.2 绘制地图

```
1 # 绘制地图
2 from pyecharts.charts import Map
3 from pyecharts import options as opts
4 data_label = "确诊"
5 map_title = "全国疫情图"
6 c = (Map() # 声明一个map对象
7     .add(data_label, map_data, "china") # 添加数据
8     # 设置标题和颜色
9     .set_global_opts(title_opts=opts.TitleOpts(title=map_title),,
10                     visualmap_opts=opts
11                         .VisualMapOpts(split_number=6, # 和自己分类的书目一致
12                                         is_pieewise=True,
13                                         # 可以根据自己需要的分类更改以下代码的最大值、最小值、标签、颜色
14                                         pieces=[{"min":1,"max":9,"label":"1-9人","color":"#ffefd7"},
15                                                  {"min":10,"max":99,"label":"10-99人","color":"#ffd2a0"},
16                                                  {"min":100,"max":499,"label":"100-499人","color":"#fe8664"},
17                                                  {"min":500,"max":999,"label":"500-999人","color":"#e64b47"},
18                                                  {"min":1000,"max":9999,"label":"1000-9999人","color":"#c91014"},
19                                                  {"min":10000,"label":"10000人及以上","color":"#9c0a0d"}
20                                         ])
21     )
```

```
22 c.render("全国实时疫情.html") # 可以自定义文件名
```

代码解析：地图绘制时，需要对以下几点做一个修改：

- 修改数据内容标签data_label;
- 传入数据map_data，数据的结构是一个列表嵌套列表或元组的格式，形如：[('地区名称1', '数据1'), ('地区名称2', '数据2'), ('地区名称3', '数据3').....]。比如：[('香港', 5169), ('上海', 1038), ('四川', 718)];
- 修改地图的主题map_title;
- 修改分类组数split_number，和pieces中的分组数量一致;
- 修改pieces中的分组界限和标签，min表示最小值，max表示最大值，label则是显示在地图上的标签；如果想要修改颜色，可以直接修改color后的16进制字符。