**Python爬虫助力疫情数据**

**摘要**

随着信息化的高速发展，大数据的引用越来越受到重视，应用也日益广泛，与此同时，数据可视化为此次新型冠状病毒肺炎疫情防控提供了有力的支持。本文旨在利用数据挖掘技术、挖掘出有关国内疫情和海外疫情的现有确诊、累计确诊、累计治愈和累计死亡等相关的数据情况，为疫情防控提供更多有效可靠信息，采用可视化工具使对疫情数据有一个更加直观了解分析，以及将可视化数据分析结果在前端进行展示。

第一，疫情相关数据的爬取与分析。本文运用Python爬虫爬取疫情相关数据并对数据进行整合。其中，爬取数据前需要获取网页的源代码，不同的数据结构都有指定的URL，根据需要选择特定的URL。获取相关数据后对数据进行解析，最后将以JSON格式存储的数据转换为字典，为后续的疫情数据可视化操作提供数据支撑。

第二，对所获取的疫情相关数据进行可视化。为了能更加深入挖掘出这些疫情数据隐藏的信息，以便对中国以及世界的疫情信息的分布以及变化趋势有更加清晰直观的了解，本文采用Pyecharts库对所获取的疫情的相关数据进行可视化，最终以图像形式、地图类型将疫情数据展示出来。其中绘制了中国各省市（区）总体疫情信息、治疗率、死亡率，中国个省市（区）总体疫情信息、世界总体信息等折线图、散点图、柱状图，以及绘制世界疫情以及地级市疫情一系列交互地图。

第三，疫情数据可视化的前端部署。本文在前端页面的部署上利用了jQuery、layui、EasyUI中现有的页面布局样式对前端页面进行美化，最终我们制作出的疫情数据可视化及前端部署网站，其链接为：'<http://106.52.252.68/>'

**关键字**：Python数据分析、Python网络爬虫、Python数据可视化

**Python crawler boosts epidemic data**

**Summary**

With the rapid development of informatization, the citation of big data is getting more and more attention, and its application is becoming more and more extensive. At the same time, data visualization provides strong support for the prevention and control of the new coronavirus pneumonia epidemic. This article aims to use data mining technology to dig out the relevant data of the existing diagnoses, cumulative diagnoses, cumulative cures and cumulative deaths related to domestic and overseas epidemics, provide more effective and reliable information for epidemic prevention and control, and use visual tools to make Have a more intuitive understanding and analysis of epidemic situation data, and display the results of visual data analysis on the front end.  
 First, crawling and analysis of epidemic-related data. This article uses the Python crawler to crawl the epidemic-related data and integrate the data. Among them, the source code of the webpage needs to be obtained before crawling the data. Different data structures have designated URLs, and select specific URLs as needed. After obtaining the relevant data, the data is parsed, and finally the data stored in JSON format is converted into a dictionary to provide data support for the subsequent epidemic data visualization operations.  
 Second, visualize the acquired epidemic-related data. In order to dig deeper into the hidden information of these epidemic data, in order to have a clearer and more direct understanding of the distribution and changing trends of epidemic information in China and the world, this article uses the Pyecharts library to visualize the acquired epidemic related data, and finally to The image form and map type will show the epidemic data. It draws the general epidemic information, treatment rate, mortality rate of various provinces and cities (districts) in China, the general epidemic information of all provinces and cities (districts) in China, the world general information and other line charts, scatter plots, histograms, and the world epidemic situation A series of interactive maps of the epidemic situation in the city.  
 Third, the front-end deployment of epidemic data visualization. This article uses the existing page layout styles in jQuery, layui, and EasyUI to beautify the front-end pages in the deployment of the front-end pages. Finally, we produced the epidemic data visualization and front-end deployment website. The link is:'http://106.52 .252.68/'

**Keywords:** natural language, Python data analysis, Python web crawler, Python data visualization

目录

[1 引言 1](#_Toc15526)

[1.1 研究背景 1](#_Toc4571)

[1.2 挖掘目标 1](#_Toc4785)

[2 数据准备（疫情相关数据） 2](#_Toc30333)

[2.1 了解Python获取网络数据的方法 2](#_Toc8481)

[2.2 Python爬取数据的流程 2](#_Toc17487)

[3 任务分析以及实现 3](#_Toc16873)

[3.1 获取疫情数据 3](#_Toc25181)

[3.1.1 获取疫情数据分析 3](#_Toc2340)

[3.1.2 获取疫情数据分析过程 3](#_Toc16534)

[3.1.3 获取疫情数据实现过程 7](#_Toc2653)

[3.1.4 获取疫情数据结果展示 9](#_Toc16991)

[3.2 疫情数据分析 10](#_Toc26949)

[3.2.1 研究疫情数据目的 10](#_Toc6915)

[3.2.2 图像分析 10](#_Toc6381)

[3.2.3 疫情数据分析实现过程 14](#_Toc4724)

[3.3 每日疫情变化可视化 17](#_Toc20306)

[3.4 中国疫情地图可视化 28](#_Toc17199)

[3.4.1 问题分析 28](#_Toc18924)

[3.4.2 可视化分析 28](#_Toc22893)

[3.4.3 实现过程 33](#_Toc8115)

[3.5 世界疫情及地级市疫情地图可视化 36](#_Toc24474)

[3.5.1 问题分析 36](#_Toc20396)

[3.5.2 时间轴图像分析 36](#_Toc25499)

[3.5.3 实现过程 43](#_Toc27021)

[4 疫情数据可视化及前端部署 51](#_Toc22885)

[4.1 页面布局设计思路 51](#_Toc5164)

[4.2 页面效果展示 52](#_Toc4824)

[4.2.1 疫情表格详情页面 52](#_Toc11903)

[4.2.2 中国疫情地图页面 52](#_Toc26230)

[4.2.3 中国地区疫情变化分析页面 53](#_Toc26619)

[4.2.4 世界疫情趋势页面 53](#_Toc18590)

[4.2.5 其他信息页面 54](#_Toc4812)

[5 总结 55](#_Toc23790)

[6 参考文献 55](#_Toc32179)

1. **引言**
   1. **研究背景**

数据新闻是一种集大数据处理以及新闻叙事为一体的新型新闻传播活动，不仅涉及数据挖掘技术，还包括数据的可视化。基于数据可视化技术的数据不再是乏味的文字数字，而是扩展到图表甚至视频、动画等，同时加上交互式的展现。比如在疫情报道中，患者所在地点查询地图、某个国家疫情确诊、治愈等情况的发展趋势，以可视化的方式呈现疫情相关数据能给读者们更多的参与。

2020年初，由于一场突如其来的疫情影响，中国全国乃至全世界进入了疫情防控的紧张时期，对于这场特殊的疫情攻坚战，疫情实时地图、感染人数建模、问卷信息收集等手段对疫情相关数据分析不仅是国家相关部门的工作，也是人们日常关心的话题之一，但是单单凭借人民日报客户端、新华网、央视新闻、微博、抖音等多媒体平台发布的数据的难以让人们直观的看到疫情总体的发展趋势，将众多庞杂的数据和新闻转变成普通人能看懂的图像内容，可以全面地展现了疫情形势。因此，我们利用数据挖掘等技术对所获取的数据进行可视化分析，让疫情相关数据以更直观、准确、快速的方式体现。

* 1. **挖掘目标**

**疫情数据获取：**爬取腾讯实时疫情网页，通过腾讯实时疫情网页可以获取的有关疫情数据的真正URL，采用Python内requests、selenium等多个模块发起用于模拟浏览器获取到网页数据的请求，可获取到如全国疫情数据、中国各省份总疫情数据、国外疫情数据等相关数据，为之后的数据分析做准备。

**疫情分析：**为了更直观地展示出中国各省份（地区）和外国各地区的疫情情况，更有效地展示出疫情从爆发到现在的发展趋势，以及关于疫情的治愈率和死亡率的情况。对爬取保存的疫情数据进行可视化，使用了Pyecharts对生成的图像进行了分析。对这些图像进行分析，获取有用于我们的关于疫情各方面情况的信息，这样有利于我们了解疫情的发展趋势，对目前疫情形势进行分析，也有利于对未来疫情发展趋势进行预估，并采取相关有用的方案进行防控。

**每日疫情可视化、中国疫情地图可视化：**为了更直观地展示出中国的疫情情况，更直观地展示中国当前确诊人数、死亡人数、治愈人数等情况，我们针对部分爬取获取的疫情数据进行数据可视化，采用了Pyecharts库，对生成的图像做进一步的分析。

**世界疫情及地级市疫情地图可视化：**使用了Pyecharts库的Bar, Line, Map, Pie绘制世界疫情以及地级市疫情一系列交互地图，以这种直观的方式，帮助我们更好的了解疫情发展变化，通过最直观清楚的确诊数据、疑似数据、治愈数据和死亡数据统计总体上了解疫情的分布趋势。

1. **数据准备（疫情相关数据）**
   1. **了解Python获取网络数据的方法**

在爬取网页数据之前，需要了解和分析网页内容的构造，根据网页构造的不同和爬取数据结构的不同，可以使用不同的Python爬虫技术对所需数据进行爬取。一般支持网页的技术是HTML/XML、AJAX和JSON，而常见的网页主要是HTML，HTML是一种超文本标记语言。“超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素。而这些元素都有指定的URL，可以通过分析URL内容来确定是否是所需要爬取数据，一般数据是以JSON进行储存的，在获取数据后，需要对数据进行解析，将JSON格式数据转换为字典。

在爬取的开始可用到的Python中相关的库，如urllib、urllib3、requests、mechanize、selenium、splinter，其中urllib、urllib3、requests、mechanize 是用来获取URL对应的原始响应内容。在解析方面可调用的库有lxml、re、beautifulsoup4、pyquery等，常用方法是xpath路径表达式、css选择器、正则表达式等，其中xpath路径表达式、css选择器主要用于提取结构化数据，而正则表达式用于提取非结构化的数据，根据不同数据类型，选择对应的方法进行爬取。

* 1. **Python爬取数据的流程**

图2-1-1 Python爬取数据的流程图

（1）爬取数据前需要获取网页的源代码，不同的数据结构都有指定的URL，根据需要选择特定的URL；

（2）基于requests模块发起用于模拟浏览器获取到网页数据的请求；

（3）获取响应中的数据；

（4）网页的数据是以JSON格式储存的，需要对数据进行解析；

（5）保存数据。

1. **任务分析以及实现**
   1. **获取疫情数据**
      1. **获取疫情数据分析**

利用Python爬取网站疫情数据，掌握当前疫情状况。打开爬取腾讯实时疫情网址找到国内疫情数据和海外疫情数据，发现网站网址不变，可以知道数据是通过ajax进行传输的，那么就可以找到数据传输的接口。查看腾讯实时疫情网页源代码，在【Network】面板下，刷新网页，找到数据接口，即可找到腾讯实时疫情网页的有关疫情数据的各个Request URL，分析这些疫情数据的资源结构，可以通过读取这些URL网页源代码获取所需要的数据，调用json.loads()将JSON格式数据转为字典类型，并将处理好的数据存储为.xlsx文件。

* + 1. **获取疫情数据分析**
       1. **数据来源**

爬取的数据来自以下的腾讯实时疫情网址，用浏览器打开本次爬取的网页并分析，<https://news.qq.com/zt2020/page/feiyan.htm#/global>该网页涉及了有关国内疫情和海外疫情的现有确诊、累计确诊、累计治愈和累计死亡的情况，数据内容详细且客观，可以通过爬取该网页来获取国内疫情数据和国外疫情数据，以及有关疫情数据的真正URL获取，初始页面以及数据获取页面如图3-1-1所示：

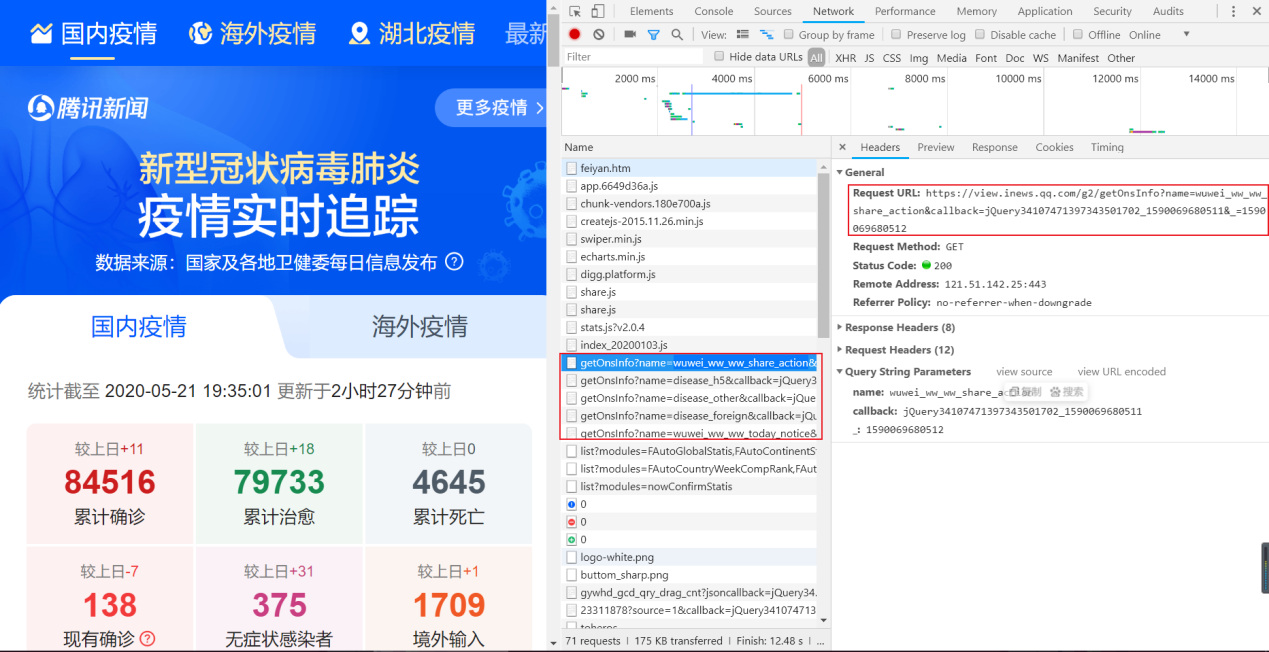


图3-1-1 URL获取

* + - 1. **Python获取网页数据任务的方法**

在爬取网页数据前需要获取该网页的源代码，通过解析源代码去获取所需要的数据。打开网页，点击【F12】，并打开【Network】面板，再点击【F5】刷新，可以从【Network】面板看到该网页的数据资源结构，而不同结构代表着不同数据，根据爬取的任务，选定对应的结构，网页结构如图3-1-2所示：

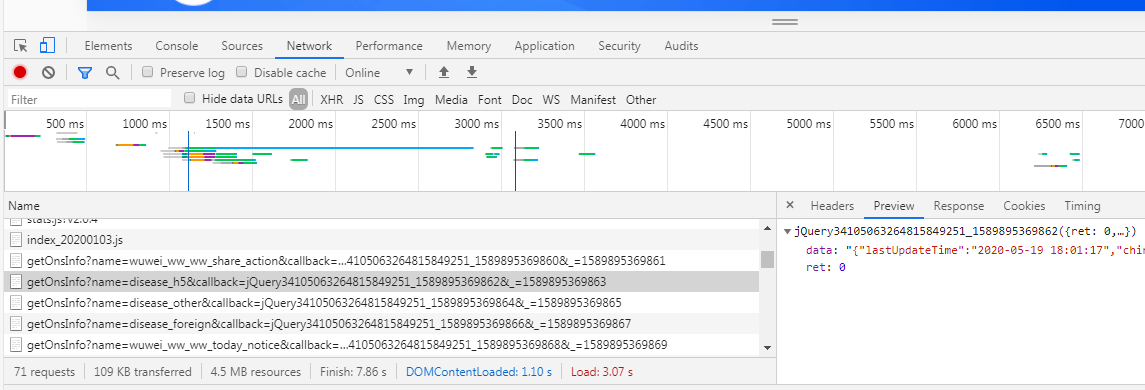


图3-1-2 网页结构

* + - 1. **爬取中国疫情和国外疫情数据**

此次任务爬取的是中国历史疫情和国外历史疫情的数据，通过在对应的结构上点击右键选择【Open link in new tab】，可以打开新的标签页，关于国内疫情数据和国外疫情数据的资源结构的内容如图3-1-3、图3-1-4所示：



图3-1-3 国内疫情数据的JSON文件预览

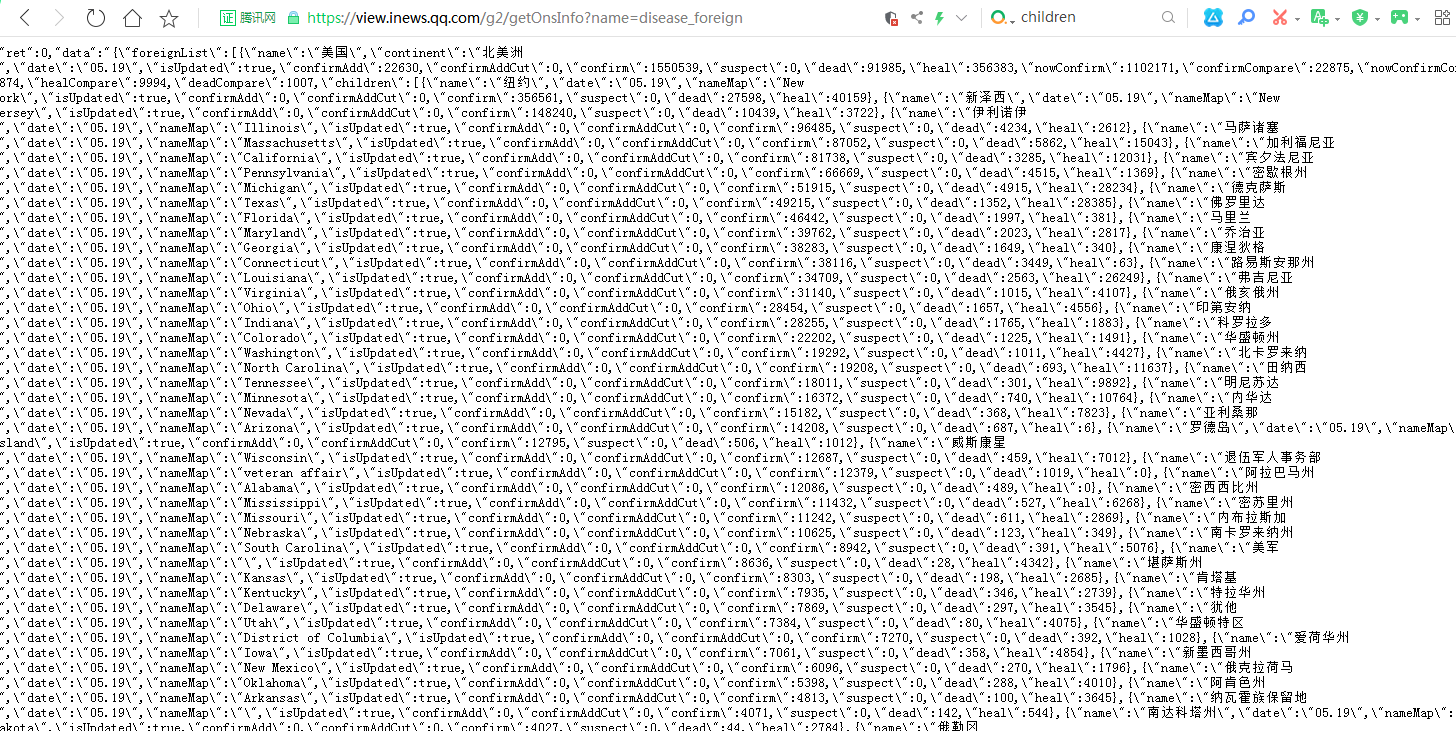


图3-1-4 国外疫情数据的JSON文件预览

通过图3-1-3、图3-1-4可以看出网页的疫情数据是以JSON格式进行储存的，可以将URL复制下来，则

关于储存中国疫情数据的URL为 self.chinatotal\_h5\_url=

"https://view.inews.qq.com/g2/getOnsInfo?name=disease\_h5"

关于储存国外疫情数据的URL为 self.foreigntotal\_url=

"https://view.inews.qq.com/g2/getOnsInfo?name=disease\_foreign"

* + - 1. **爬取中国疫情和国外疫情趋势数据**

为了了解中国疫情和国外疫情的发展趋势，需要爬取疫情趋势数据，通过在对应的结构上点击右键选择【Open link in new tab】，可以打开新的标签页，关于国内疫情趋势数据和国外趋势疫情数据的资源结构的内容如图3-1-5、图3-1-6所示：

图3-1-5 国内疫情趋势数据的JSON文件预览

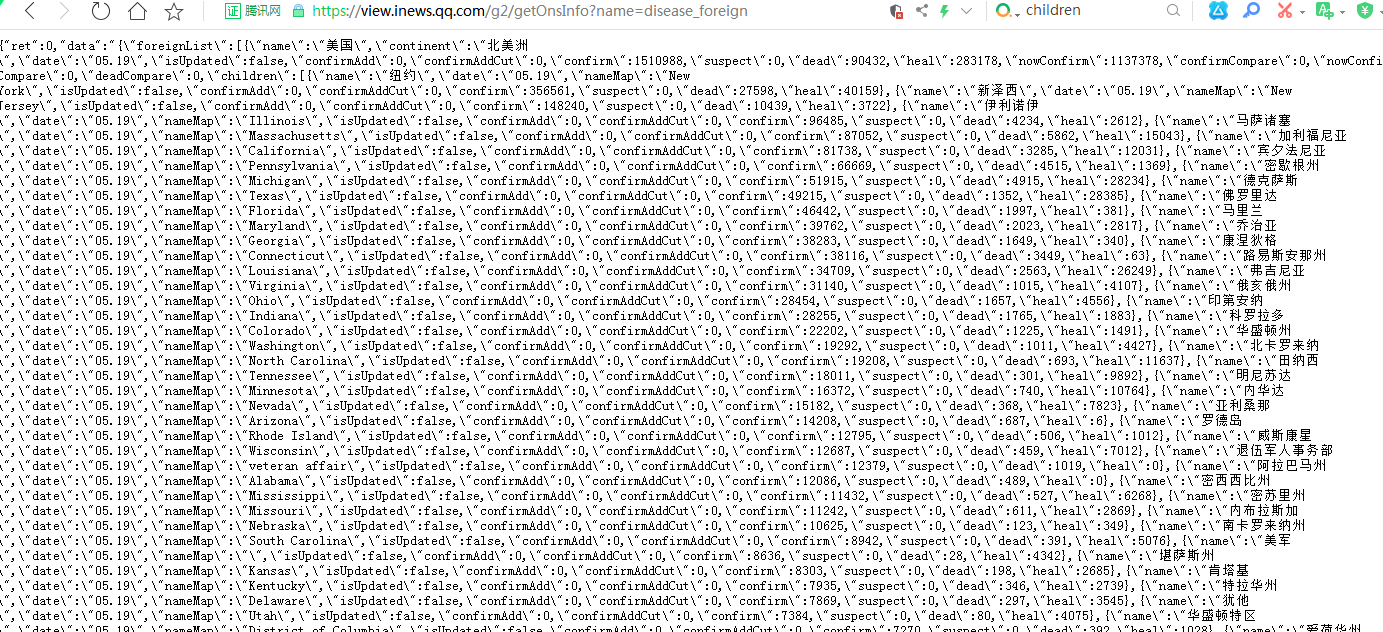


图3-1-6 国外疫情趋势数据的JSON文件预览

通过对URL内容的分析，可知关于储存国内疫情趋势数据的URL为

self.chinatotal\_url='https://view.inews.qq.com/g2/getOnsInfo?name=disease\_other'；

关于储存国外疫情趋势数据的URL为

self.foreigntotal\_country\_url=’https://api.inews.qq.com/newsqa/v1/automation/foreign/daily/list?country=’

* + - 1. **爬取中国各省市（区）历史疫情数据**

为了更直观地了解中国各省市（区）历史疫情数据情况，在对应的结构上点击右键选择【Open link in new tab】，可以打开新的标签页，关于中国各省市（区）历史疫情数的内容如图3-1-7所示：



图3-1-7 中国各省市（区）历史疫情数据

通过对URL内容的分析，可知关于中国各省市（区）历史疫情数据的URL为

self.provincetotal\_url='https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?province='

* + 1. **获取疫情数据实现过程**
       1. **获取实时疫情页面网页源码并解析JSON数据**

首先调用requests.get()实现HTTP请求，并调用json.loads()将JSON格式数据转为字典类型，实现的部分代码3-1-1所示：

代码3-1-1 获取网页

**class** **getCovidData(**object**):**

**def** \_\_init\_\_**(**self**):**

# 全国疫情数据

self**.**chinatotal\_h5\_url **=** "https://view.inews.qq.com/g2/getOnsInfo?name=disease\_h5"

# 国内疫情趋势数据

self**.**chinatotal\_url **=** 'https://view.inews.qq.com/g2/getOnsInfo?name=disease\_other'

# 省区信息请求网址

**def** getChinaAbsData**(**self**):**

response **=** requests**.**get**(**self**.**chinatotal\_h5\_url**).**json**()**

data **=** json**.**loads**(**response**[**'data'**])**

**return** data**[**'chinaTotal'**]**

* + - 1. **爬取国内疫情数据**

根据构造好的URL集，进而提取国内疫情数据，部分代码如代码3-1-2所示：

代码3-1-2 爬取国内疫情数据

**class** **ChinaData():**

**def** \_\_init\_\_**(**self**):**

self**.**coviddata **=** getData**.**getCovidData**()**

self**.**all\_data **=** self**.**coviddata**.**getChinaTotalData**()**

self**.**abs\_data **=** self**.**coviddata**.**getChinaAbsData**()**

# 获取从1月13日起的一系列累计数据

file**.**save**(**'./data/中国总体历史疫情信息/历史总体信息.xlsx'**)**

everyday\_infect**.**append**(**int**(**everyday**[**'infect'**]))**

everyday\_deadRate**.**append**(**float**(**everyday**[**'deadRate'**]))**

everyday\_healRate**.**append**(**float**(**everyday**[**'healRate'**]))**

* + - 1. **爬取国外疫情数据**

可针对构造好的URL集爬取国外疫情数据,部分代码如代码3-1-3所示：

代码3-1-3 爬取国外疫情数据

**class** **ForeignData():**

**def** \_\_init\_\_**(**self**):**

self**.**coviddata **=** getData**.**getCovidData**()**

self**.**all\_data **=** self**.**coviddata**.**getForeignTotalData**()**

# 世界总体历史疫情信息

**def** foreign\_total\_data**(**self**):**

foreign\_data **=** self**.**all\_data

foreign\_name **=** list**()**

foreign\_total\_continent **=** list**()**

foreign\_total\_isUpdated **=** list**()**

**for** country **in** foreign\_data**:**

foreign\_name**.**append**(**country**[**'name'**])** # 国家名字

foreign\_total\_continent**.**append**(**country**[**'continent'**])**

foreign\_total\_isUpdated**.**append**(**country**[**'isUpdated'**])**

foreign\_total\_confirmAdd**.**append**(**int**(**country**[**'confirmAdd'**]))**

file**.**save**(**'./data/各国各地区疫情信息/' **+** country**[**'name'**]** **+** '.xlsx'**)**

**def** main**(**self**):**

self**.**foreign\_total\_data**()**

self**.**foreign\_children\_data**()**

**if** \_\_name\_\_ **==** '\_\_main\_\_'**:**

world\_data **=** ForeignData**()**

world\_data**.**main**()**

* + - 1. **爬取国外疫情趋势数据**

可以针对构造好的URL集爬取国外疫情趋势数据,部分代码如代码3-1-4所示。

代码3-1-4 爬取外国疫情趋势数据

  # 中国这个国家的疫情数据

**def** getChinaAbsData**(**self**):**

response **=** requests**.**get**(**self**.**chinatotal\_h5\_url**).**json**()**

data **=** json**.**loads**(**response**[**'data'**])**

**return** data**[**'chinaTotal'**]**

# 中国各省份总疫情数据

**def** getProvinceCity**(**self**):**

# 发出请求并 json化处理

response **=** requests**.**get**(**self**.**province\_city\_url**).**json**()**

data **=** json**.**loads**(**response**[**'data'**])**

province **=** list**()**

city **=** collections**.**defaultdict**(**list**)**

**for** i **in** range**(**len**(**data**)):**

province**.**append**(**data**[**i**][**'province'**])**

city**[**data**[**i**][**'province'**]].**append**(**data**[**i**][**'city'**])**

**return** province**,** city

# 国内疫情数据

**def** getForeignTrendData**(**self**,** foreign\_name**):**

response **=** requests**.**get**(**self**.**foreigntotal\_country\_url **+** foreign\_name**).**json**()**

data **=** response**[**'data'**]**

**return** data

* + 1. **获取疫情数据结果展示**

各国各地区部分疫情信息以及各国历史部分疫情信息如图3-1-8所示：



图3-1-8 疫情信息文件

中国各省的城市历史部分疫情信息以及中国总体历史疫情信息如图3-1-9所示：



图3-1-9 中国疫情信息文件

部分数据内容展示如图3-1-10所示：

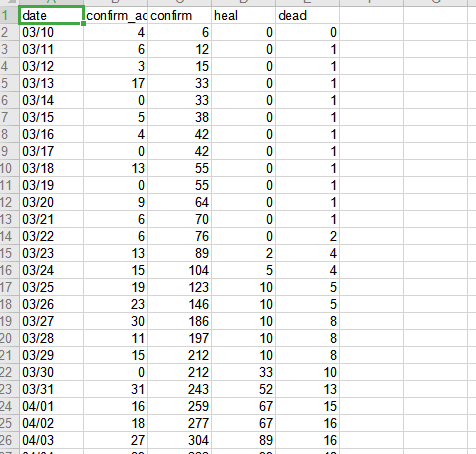
****

图3-1-10 部分数据内容

* 1. **疫情数据分析**
     1. **研究疫情数据目的**

为了更直观地展示出中国各省份（地区）和外国各地区的疫情情况，更有效地展示出疫情从开始到爆发再到平缓的过程，以及关于疫情的治愈率和死亡率的情况，我们针对部分疫情数据进行可视化，使用了Pyecharts库，对生成的图像进行了分析，并新建了文件夹analysis用于存放所生成的所有图像文件。

根据对数据的统计、分类和分析，我们绘制出了“中国每日新增信息”、“中国每日治疗率、死亡率”、“中国个省市（区）总体疫情信息”、“中国各省市(区)治疗率、死亡率”、“中国目前疫情信息”、“中国目前治疗率、死亡率”“世界总体疫情信息”以及“世界其他信息”的散点图。对这些图像进行分析，获取有用于我们的关于疫情各方面情况的信息，这样有利于我们了解疫情的发展趋势，对目前疫情形势进行分析，也有利于对未来疫情发展趋势进行预估，并采取相关有用的方案进行防控。

* + 1. **图像分析**

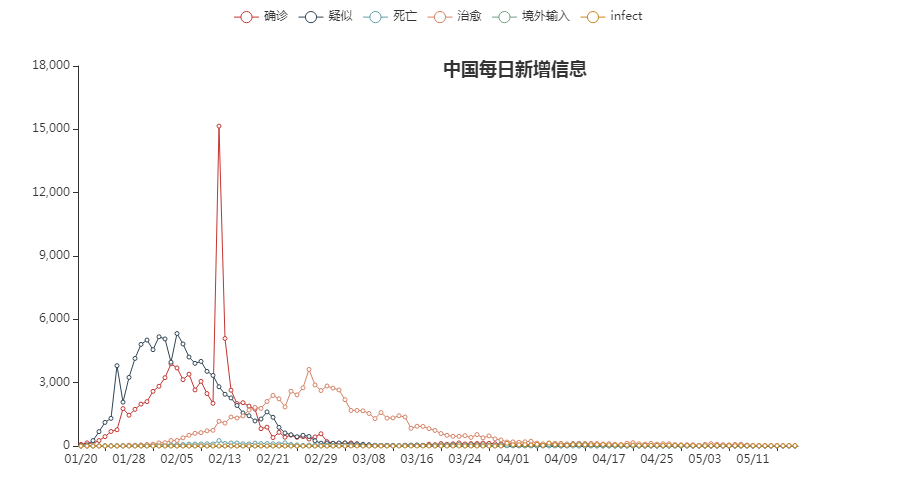
****

图3-2-1 中国每日新增确诊

从图3-2-1中可看出在2月12日，中国新增确诊人数(confirm)高达一万五千以上，之所以有这么大的差距，是因为检测新冠肺炎的方式发生了变化以及对新冠肺炎确诊的排查范围进行了扩大。检测新冠肺炎的方法新增了核磁CT，此方法从检测到确定所花费的时间上得到了缩短，有利于对新冠肺炎患者进行有效地排查。高峰期的到来表明冠状病毒隐形载体传播者的大部分问题已经被逐一发现。从另一方面可以看出这种流行病的预防和控制起到了一定的作用，也明确了日后疫情防控的方向。

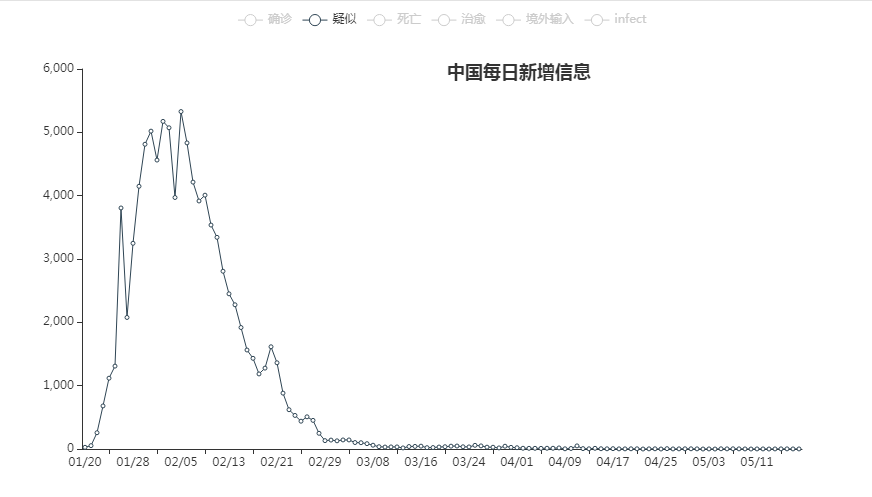


图3-2-2 中国每日新增疑似病例

从图3-2-2可以看出1月26日到2月5日的疑似病例在增加，反映出了疫情爆发的开始检测新冠肺炎的范围不大和方法不够有效。而之后疑似病例的减少也意味着关于疫情防控加强了重视。

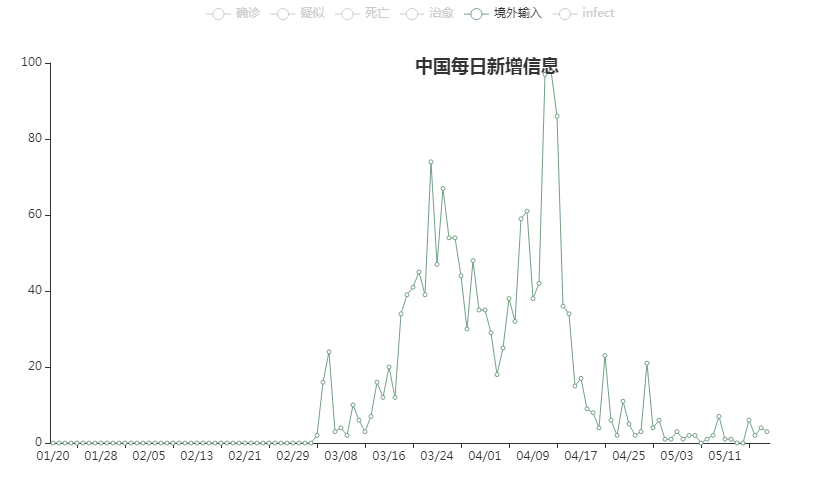


图3-2-3 中国每日新增境外输入

从图3-2-3可以看出从3月4日起，我国的新增境外输入确诊患者人数持续增加，这其中伴随着的原因是国外疫情严峻形势的不断加剧，除了对国内疫情进行防控外，还需要加强对境外输入的严控。



图3-2-4 中国每日死亡情况

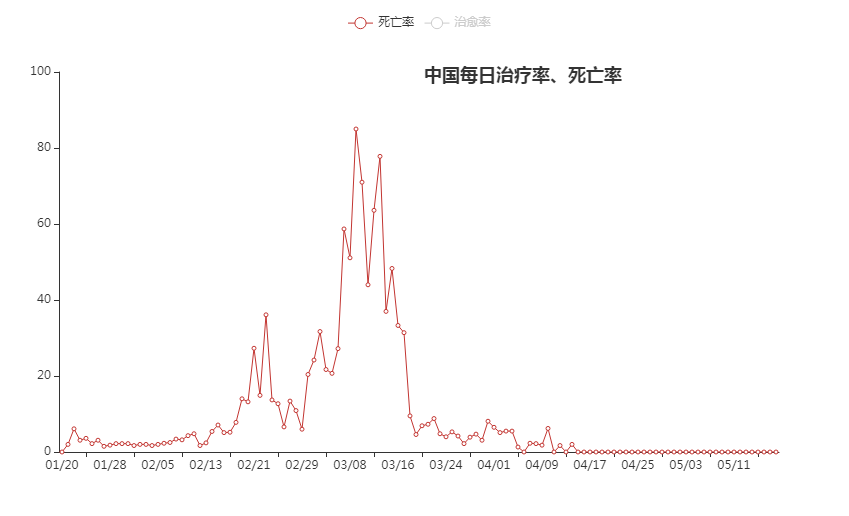


图3-2-5中国每日死亡率

从图3-2-4、图3-2-5可以看出，疫情爆发的前期，由于人们的防范意识不强，对疫情形势的严峻程度不了解，以及医疗防护用具和医疗人员的缺少，让新冠肺炎患者错失了及时确诊并及时治疗的最佳时机，以致于死亡人数增多了。



图3-2-6 中国每日治愈情况

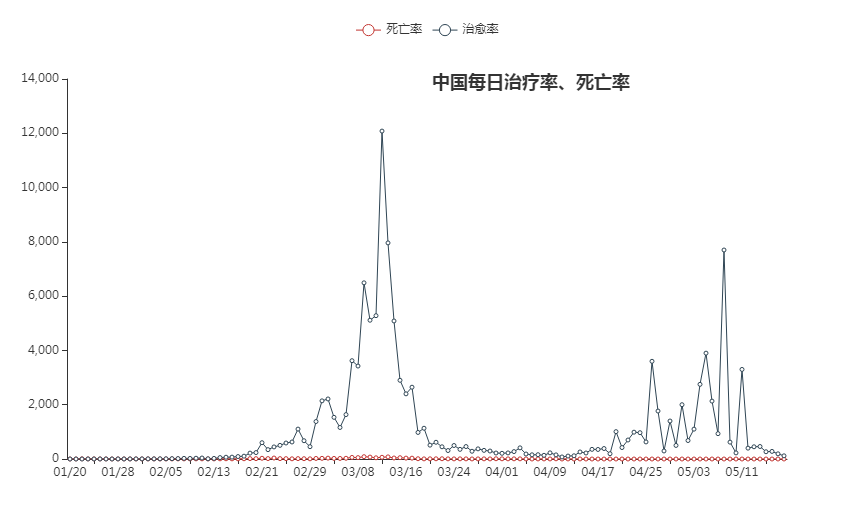
****

图3-2-7 中国每日治愈率

从图3-2-6、图3-2-7可以看出，随着国家和人们的重视，以及医护人员异地支援和防护用具的补缺，新冠肺炎的治愈率有所提高，从2月24日之前，中国的治愈人数不断增多，意味着疫情防控的不断加强，而之后治愈人数的减少也反映出了对疫情的防控是稳定且有效的。

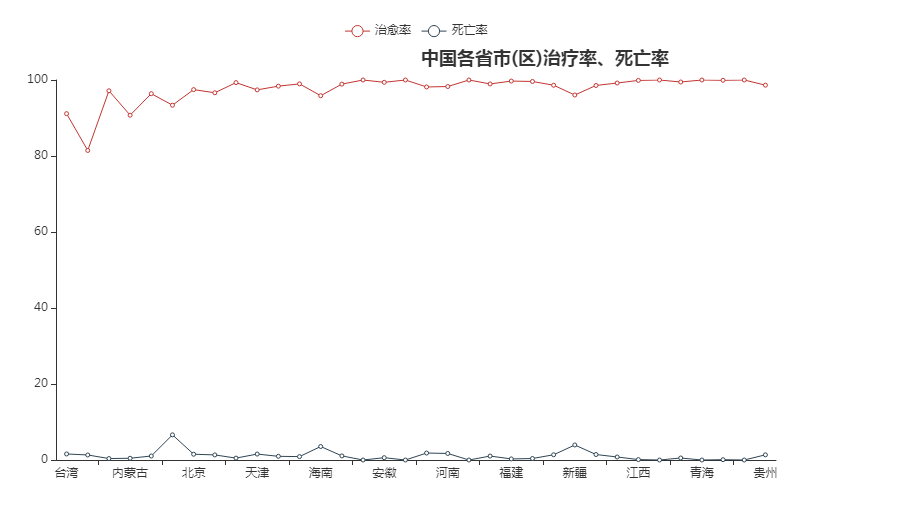


图3-2-8 中国个省市（区）治愈率和死亡率

从图3-2-8可以看出，中国各省份地区的治愈率除了吉林地区以外都在90%以上，从总体上看，国内在疫情防控上的效果明显，但也有部分地区的防控能力有待加强。

* + 1. **疫情数据分析实现过程**

**导入库、读取数据**

代码 3-2-1

**import** pandas **as** pd

**import** os

**from** pyecharts**.**charts **import** Line#折线图绘制所需要的库

**from** pyecharts**.**charts **import** Page#将所有图像放在同一页所需要的库

**import** pyecharts**.**options **as** opts#设置全局变量

file**=**os**.**getcwd**()**#读取当前文件路径

path**=**file**+**'\\analisis'

isExists**=**os**.**path**.**exists**(**path**)**#判断当前目录是否有文件夹analysis，如果没有则创建

**if** **not** isExists**:**

os**.**mkdir**(**path**)**#在当前路径创建新文件夹analisis,用于存放生成的图像数据

page**=**Page**()**#创建一个分页用于放所有的图像在这一分页上

**中国每日新增信息**

代码 3-2-2

**class** **Analyse():**

**def** analyse\_Chinaeveryday**(**self**):**

Chian\_day**=**pd**.**DataFrame**(**pd**.**read\_excel**(**file**+**'\\data\\中国总体历史疫情信息\\历史每日新增信息.xlsx'**))**#读取信息

CNday\_date**=**list**(**Chian\_day**[**'date'**])**

CNday\_confirm**=**list**(**Chian\_day**[**'confirm'**])**

CNday\_suspect**=**list**(**Chian\_day**[**'suspect'**])**

CNday\_dead**=**list**(**Chian\_day**[**'dead'**])**

CNday\_heal**=**list**(**Chian\_day**[**'heal'**])**

CNday\_importedCase**=**list**(**Chian\_day**[**'importedCase'**])**

CNday\_infect**=**list**(**Chian\_day**[**'infect'**])**

CNday\_deadRate**=**list**(**Chian\_day**[**'deadRate'**])**

CNday\_healRate**=**list**(**Chian\_day**[**'healRate'**])**

line\_CNday**=**\

**(**

Line**()**

**.**add\_xaxis**(**xaxis\_data**=**CNday\_date**)**#xaxis\_data为x轴，y\_axis为y轴

**.**add\_yaxis**(**series\_name**=**'确诊'**,**y\_axis**=**CNday\_confirm**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**add\_yaxis**(**series\_name**=**'疑似'**,**y\_axis**=**CNday\_suspect**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**#数据值得不显示

**.**add\_yaxis**(**series\_name**=** '死亡'**,**y\_axis**=**CNday\_dead**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**add\_yaxis**(**series\_name**=**'治愈'**,**y\_axis**=**CNday\_heal**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**add\_yaxis**(**series\_name**=**'境外输入'**,**y\_axis**=**CNday\_importedCase**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**add\_yaxis**(**series\_name**=**'infect'**,**y\_axis**=**CNday\_infect**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**'中国每日新增信息'**,**pos\_left**=**'50%'**,**pos\_top**=**'10%'**),**tooltip\_opts**=**opts**.**TooltipOpts**(**axis\_pointer\_type**=** 'cross'**)))**#创建折线图

**中国每日治疗率、死亡率**

代码 3-2-3

line\_CNday\_rate**=**\

**(**

Line**()**

**.**add\_xaxis**(**xaxis\_data**=**CNday\_date**)**

**.**add\_yaxis**(**series\_name**=**'死亡率'**,**y\_axis**=**CNday\_deadRate**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**add\_yaxis**(**series\_name**=**'治愈率'**,**y\_axis**=**CNday\_healRate**,**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**'中国每日治疗率、死亡率'**,**pos\_left**=**'50%'**,**pos\_top**=**'10%'**),**tooltip\_opts**=**opts**.**TooltipOpts**(**axis\_pointer\_type**=** 'cross'**))**

**)**#创建折线图

**图像保存**

代码 3-2-4

line\_CNhis**.**render**(**path**+**'\\Chinahistory.html'**)**#将生成的图像保存为html

并存放在analisis文件夹

line\_CNhis\_**.**render**(**path**+**'\\Chinahistoryrate.html'**)**#将生成的图像保存为html文件并存放在analisis文件夹

page**.**add**(**line\_CNhis**)**#将rate图像加入同一页

page**.**add**(**line\_CNhis\_rate**)**#将图像加入同一页

**def** renderall**(**self**):**

page**.**render**(**path**+**'\\analyse\_all.html'**)**#生成包含所有图像的html并存放在analysis文件夹

**def** main**(**self**):**

self**.**analyse\_Chinaeveryday**()**#生成中国每日新增信息的折线图

self**.**analyse\_province**()**#生成中国各省份信息折线图

self**.**analyse\_worldhistory**()**#生成世界总体信息折线图

self**.**analyse\_Chinahistory**()**#生成中国总体信息折线图

self**.**renderall**()**

**if** \_\_name\_\_**==**'\_\_main\_\_'**:**

analysis**=**Analyse**()**

analysis**.**main**()**

* 1. **每日疫情变化可视化**
     1. **问题分析**

利用爬取的疫情数据，将可视化数据分析结果在前端展示。运用pyecharts对中国各省境外输入汇总、中国各省市（区）历史疫情信息、中国各省的城市历史疫情情况、中国总体疫情数据、各国历史疫情数据、中国各地区疫情信息、整体疫情信息、疫情严重程度排名前十国家的数据进行详细分析、生成.HTML文件展示。

对疫情数据进行可视化后可以直观地了解到疫情发展的趋势，这对于疫情的防控、治疗、科学研究等都具有可观的作用。我们对每个国家，每个城市都进行了可视化，特别对确诊人数、新增感染人数、治愈人数、死亡人数等进行可视化，并重点中国各城市以及对一些重要的国家疫情进行详细分析。以便对中国以及世界的疫情信息有个大致的了解。这对于防控措施，防控手段都具有科学意义。

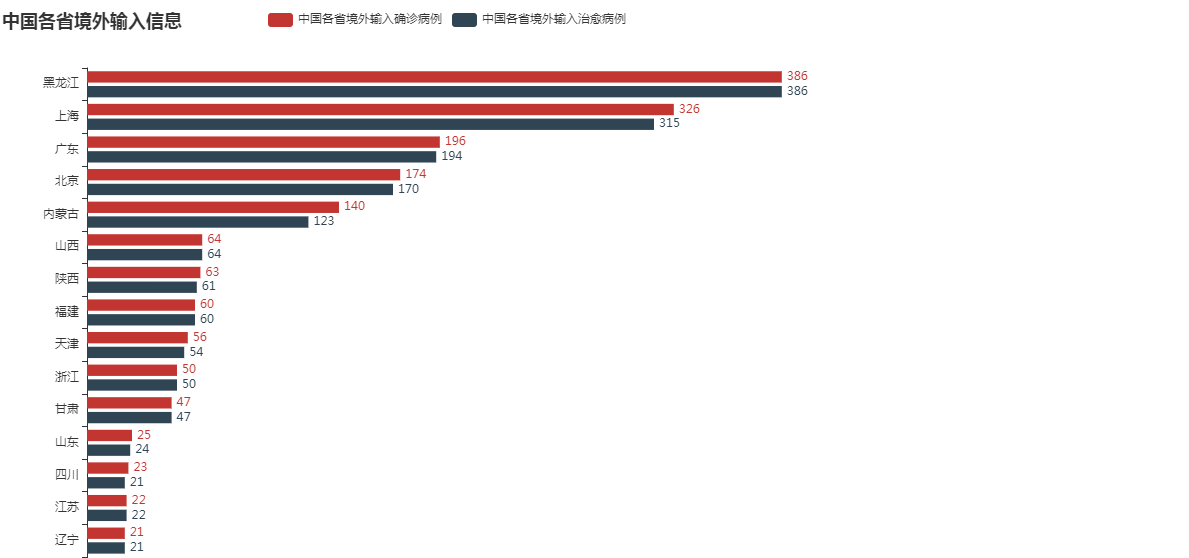
* + 1. **可视化分析与实现过程**
       1. **中国疫情总体信息**

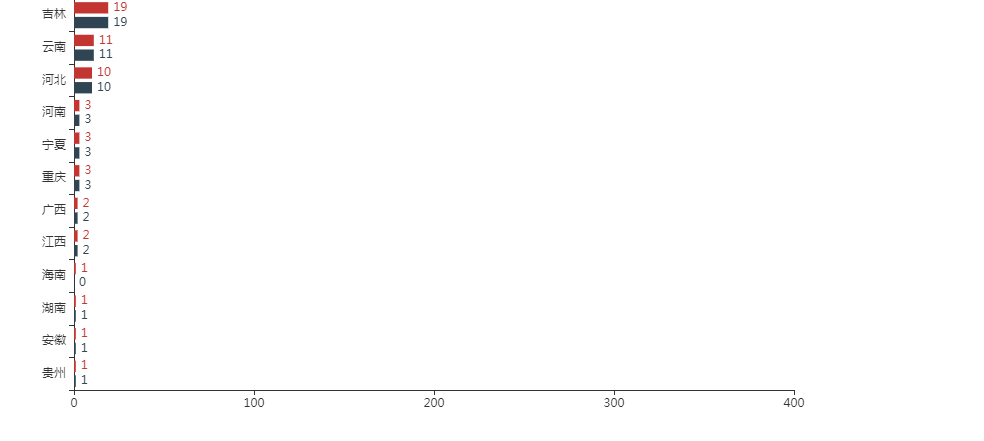
图3-3-1 中国疫情总体信息

我们由图可以看出中国疫情历史总体信息类似于“S”型曲线，从发现第一例到慢慢地爆发，再到后来我们国家大致控制住疫情。虽然前期虽然我们的防控难度大，但我们的力度也大。现在我国的疫情已经基本控制住，接下来的工作就是复工复产复学以及防控境外输入。

* + - 1. **中国各省境外输入汇总**

运用柱状图对中国各省境外输入进行可视化，然后对直观展示的数据进行分析，数据图如图3-3-2所示：



图3-3-2 中国各省境外输入汇总

由图中数据可以直观地发现，境外输入的数量大得惊人，因此对于防境外输入我们一点也不能松懈，由于黑龙江省以及内蒙古防控不够重视外，其他省份的境外相对好一点，而北京、上海、广东的境外输入多主要是因为国际航班的原因，目前已经限制部分国际航班通航，因此，日后境外输入也会得到缓解。

* + - 1. **中国各省市（区）历史疫情信息（武汉、吉林为例）**

接着运用更加可观的折线图对中国各省市（区）历史疫情信息进行可视化，可以清晰地看清楚每日的确诊人数变化、治愈人数的增加等，因此更加具有可分析性，我们以疫情比较严重的武汉、吉林为例，数据展示如图3-3-3、图3-3-4所示：

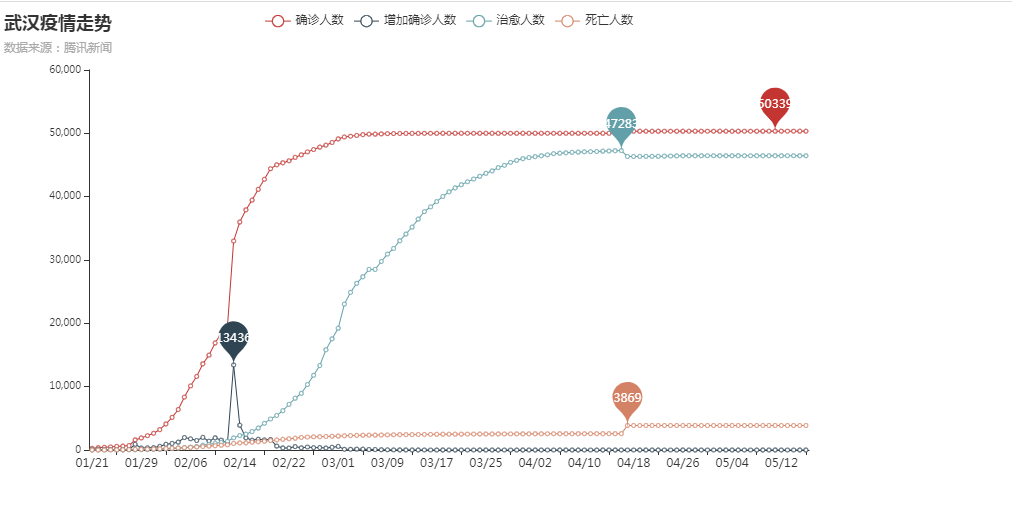


图3-3-3 中国各省市（区）历史疫情（武汉）

我们可以发现现在的武汉疫情情况经过艰苦卓绝的努力，付出巨大的代价，每天确诊人数趋于平稳，说明现已经大致已经趋于稳定，说明武汉保卫战取得了阶段性胜利，接下里就是要防治再次爆发。

图3-3-4 中国各省市（区）历史疫情（吉林）

由于吉林在后来的防控疫情中出现了松弛，导致集体感染人数较多，使人们变得人心惶惶，而由图我们也可以发现5月份吉林的确诊人数直线上升，由此知道，在疫情防控中我们不能出现松懈、麻痹的心里。

* + - 1. **中国各省市历史疫情信息（湖北、北京为例）**

毫无疑问疫情防控要具体到市区甚至到镇，各省市的总体疫情情况汇总对于各省的具体防控措施的实施起到决定性作用，因此，我们要对全国各省城市的历史疫情也进行可视化，然后进行分析，同样地我们运用折线图对中国各省的城市历史疫情信息进行可视化，我们以湖北、以及北京为例，效果如图3-3-5、3-3-6示：

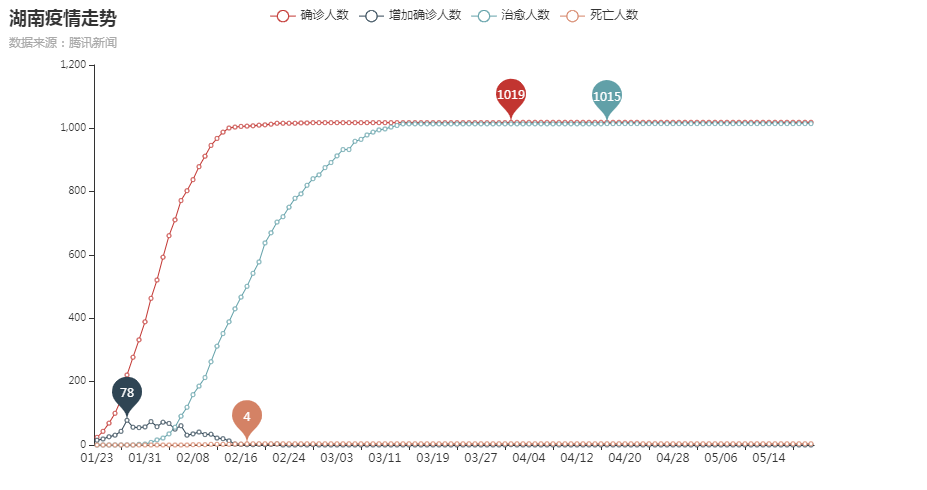
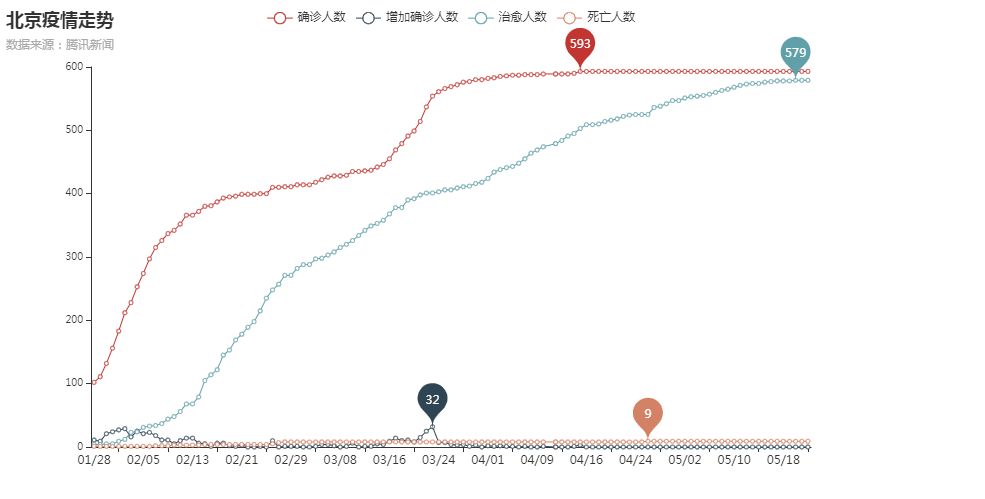


图3-3-5 中国各省市历史疫情信息（湖北省）

我们发现在3月份以后，湖北的确诊人数呈现大幅度下降，再到后来的趋于0，这说明只要武汉保卫战取得胜利，湖北保卫战就一定能取到胜利。

图3-3-6中国各省市历史疫情信息（北京市）

由于北京市是我们的首都，经济中心，文化中心等，因此，人口流动性非常大，而前期由于不知道疫情的传染性，导致前期确诊人数大幅度上升，经过治疗、防控，现在我们可以看到确诊人数已经平稳，取到阶段性胜利，到由于境外输入非常严重，因此，防控境外输入是一下的重要工作。

* + - 1. **各国历史疫情信息（美国、塞尔维亚为例）**

中共十八大明确提出要倡导“人类命运共同体”意识。习近平就任总书记后首次会见外国人士就表示，国际社会日益成为一个你中有我、我中有你的“命运共同体”。而且疫情没有国界，海外对于疫情防控形势非常严峻，因此，我们也对海外各国各地区的疫情信息运用折线图进行可视化，以便对于研究分析，我们以美国以及我们早期派专家组去帮助的塞尔维亚为例说明，数据如图3-3-7、图3-3-8所示：

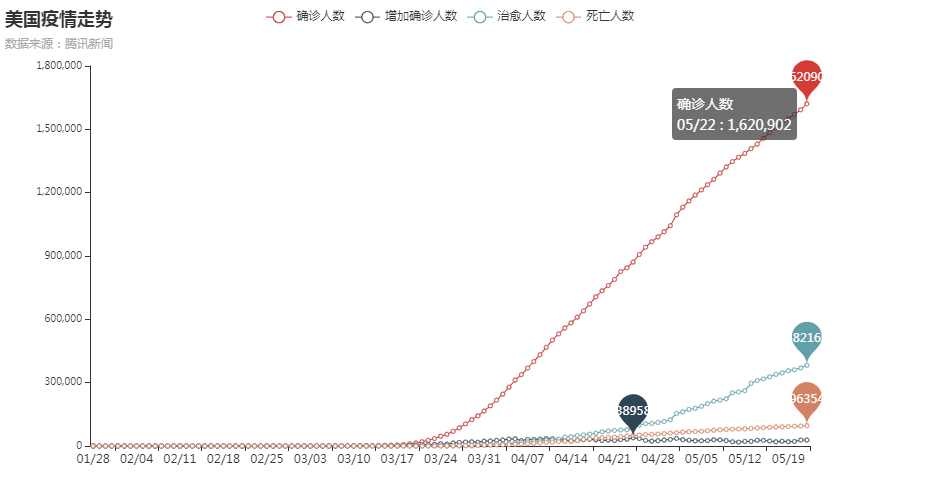


图3-3-7 各国历史疫情信息（美国）

由于美国前期不重视疫情的防控，导致浪费中国为他们争取抗疫情的宝贵时间，后来美国出现大规模的爆发，集体传染，导致一发不可收拾，现在美国的确诊人数已经达到一百五十多万，而且每天还以巨大的数量上升，这也说明对待疫情我们不能有任何的轻视心态。

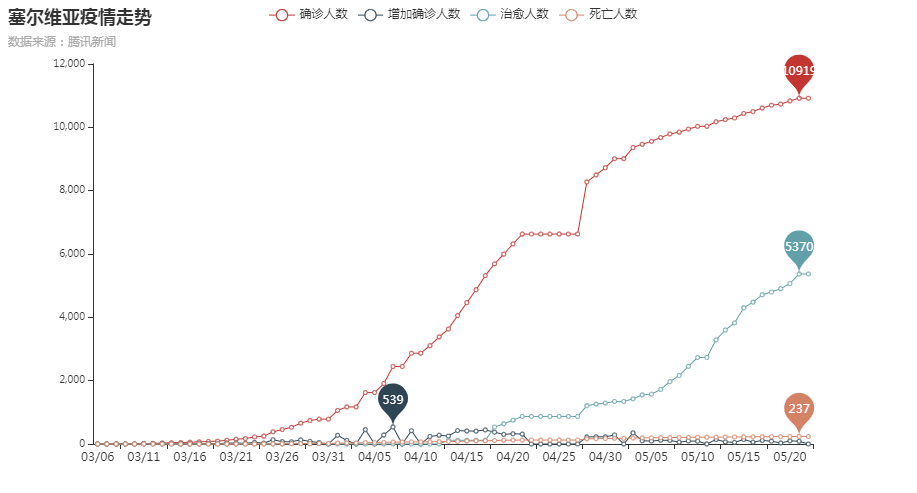


图3-3-8 各国历史疫情信息（塞尔维亚）

虽然塞尔维亚前期的疫情也异常严重，但他们却很重视，丝毫不敢放松，请求中国派专家组以及支援抗疫物资，而现在经过努力，塞尔维亚的疫情也已经趋于平稳，得到了可控。尽管如此，他们也丝毫不敢放松，继续加大防控力度。

* + - 1. **各大洲的疫情信息**

而了解完各国个地区的疫情情况后，我们也对各大洲疫情的数据运用饼图进行可视化，因为这样可以了解更多情况，这对于外贸、航线开通等具有重要意义。，下面以各大洲的疫情信息，如图3-3-9：

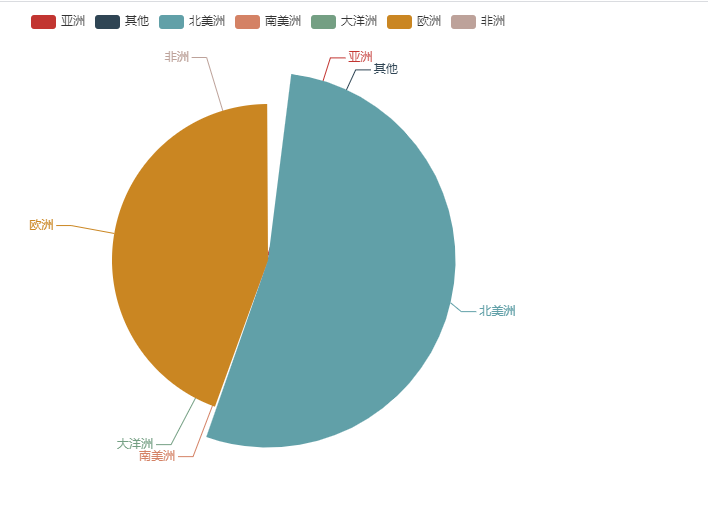


图3-3-9 各大洲疫情信息

虽然前期亚洲的疫情比较严峻，但经过防控力度的加大，现在亚洲的疫情也得到很好的控制，而由于美国所在的北美洲的疫情形势显然不是很乐观，因此对于大洋洲的航班，我们应该采取多方面的研究才能通航。而且对于入境人员也应该进行严重的隔离。

我们再以各大洲的现存人数以及治愈信息来了解更多信息，具体见图3-

3-10,图3-3-11:

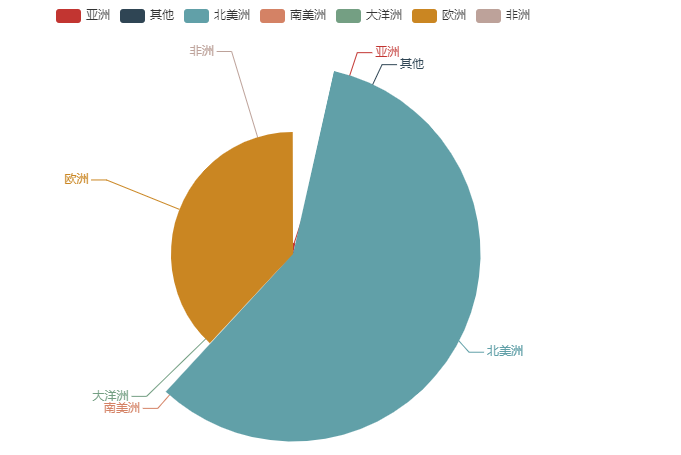


图3-3-10 各大洲治愈信息

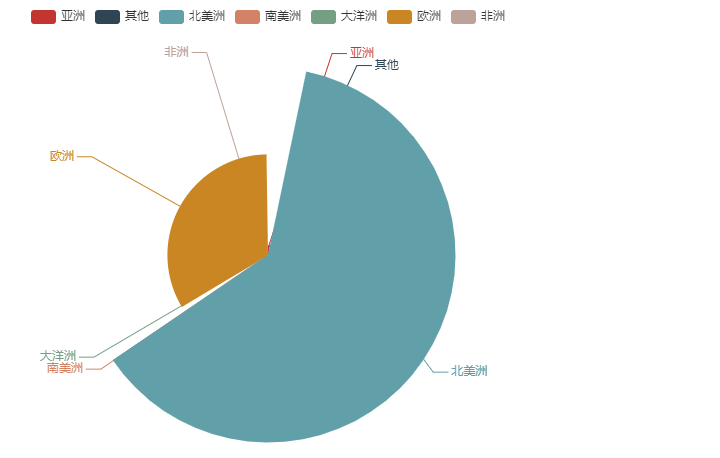


图3-3-11 各大洲现存确诊信息

由可视化图像发现，虽然北美洲的治愈信息比较好，但是他们的现存确诊信息也是比较多，而非洲国家的医疗水平比较差，治愈率相对较低，因此，世界卫生组织以及一些国家应该加大对非洲国家的帮助。

* + - 1. **疫情严重程度前十的国家**

海外部分国家的疫情非常严重，他们的控制疫情的情况对于全世界防控疫情影响非常大，因此，我们也对全球疫情最严重的前十个国家运用折线图以及柱状图进行可视化，以便了解他们控制疫情的情况，展示效果见图3-3-12、图3-3-13：

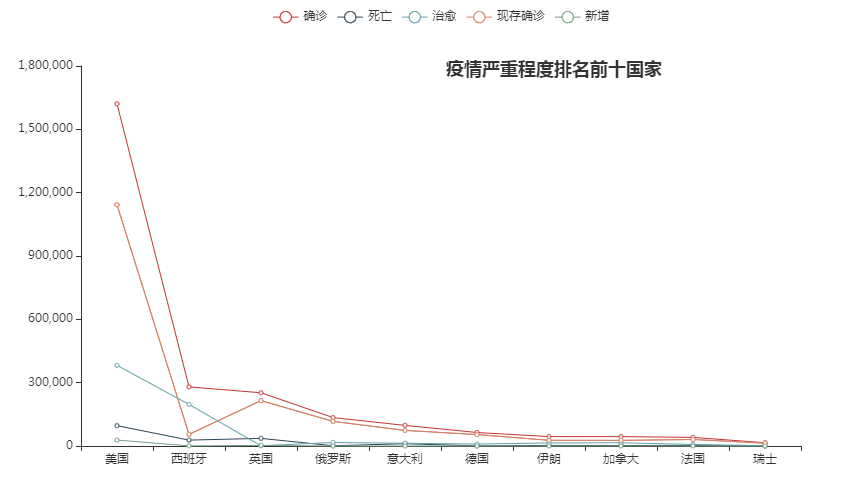
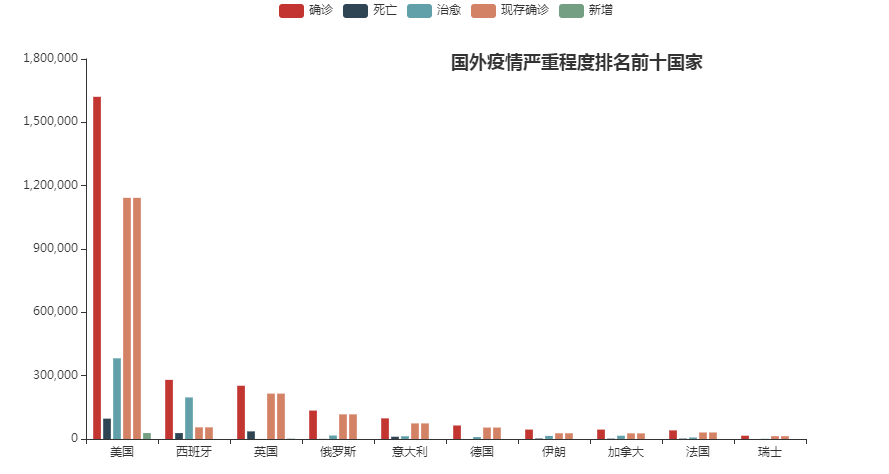


图3-3-12 疫情严重程度排名前十的国家（折线图）

图3-3-13 疫情严重程度排名前十的国家（柱状图）

由图3-3-12、图3-3-13我们可以清晰的发现，美国的疫情现在已经非常严重，必须采取一系列严格的措施的来防控。全球疫情严重程度前十的国家的总确诊人数都十分惊人，疫情没有国界，是全人类的公敌，因此，我们应该携起手来共同抗击疫情，树立人类命运共同体的理念。对于获取的疫情可视化后，我们可以非常清楚地看到中国各省市（区），以及全球等的疫情信息，这对于每天疫情的防控都起到可靠的依据，世界因每天的 疫情数字变得不一样。

* + 1. **每日疫情变化可视化实现过程**

**导入相关信息：**

代码3-3-1

readPath **=** dwf**.**rootPath #读取数据的地方

savaPath **=** dwf**.**savaPathAnalyse #生成结果的地方

flag **=** dwf**.**flag #是否生成的数据

**中国各省境外输入汇总：**

代码3-3-2

# 显示所有列

pd**.**set\_option**(**'display.max\_columns'**,** **None)**

china\_imported **=** a**.**drop\_duplicates**(**subset**=[**'name'**],** keep**=**'first'**)**

china\_imported**.**index **=** range**(**len**(**china\_imported**))**

china\_imported **=** china\_imported**.**sort\_values**(**axis**=**0**,** ascending**=True,** by**=[**'confirm'**])**

# 将dataframe格式转换成相应的列表格式

china\_imported\_name **=** china\_imported**[**'name'**]**

china\_imported\_name **=** list**(**china\_imported\_name**)**

china\_imported\_confirm **=** china\_imported**[**'confirm'**]**

china\_imported\_confirm **=** list**(**china\_imported\_confirm**)**

china\_imported\_heal **=** china\_imported**[**'heal'**]**

china\_imported\_heal **=** list**(**china\_imported\_heal**)**

**各国各地区疫情信息：**

代码3-3-3

**def** foregin\_city\_all**():**

file **=** os**.**getcwd**()** # 读取当前文件路径

path **=** file **+** '\\data\\各国各地区疫情信息'

path1 **=** file **+** '\\daliy\_changes\_view'

isExists **=** os**.**path**.**exists**(**path1**)** # 判断当前目录是否有文件夹photos，如果没有则创建

**if** **not** isExists**:**

os**.**mkdir**(**path1**)** # 创建photos,用于存放生成的图像数据

path2 **=** file **+** '\\daliy\_changes\_view\\各国疫情严重程度排名前十地区信息'

isExists **=** os**.**path**.**exists**(**path2**)** # 判断当前目录是否有文件夹daliy\_changes\_view，如果没有则创建

**if** **not** isExists**:**

os**.**mkdir**(**path2**)** # 创建daliy\_changes\_view,用于存放生成的图像数据

Filelist **=** **[]** # 将当前文件夹内的所有表名存放此列表

**for** home**,** dirs**,** files **in** os**.**walk**(**path**):**

**for** filename **in** files**:**

Filelist**.**append**(**filename**)**

# 判断数据是否存在

**def** checknan**(**name**):**

**if** np**.**any**(**pd**.**isnull**(**name**))** **==** **True:**

name**.**fillna**(**value**=**"0"**,** inplace**=True)**

# 随机颜色生成用于制作南丁格尔玫瑰图

**def** randomcolor**(**kind**):**

colors **=** **[]**

**for** i **in** range**(**kind**):**

colArr **=** **[**'1'**,** '2'**,** '3'**,** '4'**,** '5'**,** '6'**,** '7'**,** '8'**,** '9'**,** 'A'**,** 'B'**,** 'C'**,** 'D'**,** 'E'**,** 'F'**]**

color **=** ""

**for** i **in** range**(**6**):**

color **+=** colArr**[**random**.**randint**(**0**,** 14**)]**

colors**.**append**(**"#" **+** color**)**

**return** colors

**for** i **in** Filelist**:**

data **=** pd**.**read\_excel**(**file **+** '\\data\\各国各地区疫情信息\\%s' **%** **(**i**),** index**=False)**

data\_sort **=** data**.**sort\_values**(**axis**=**0**,** ascending**=False,** by**=[**'confirm'**])**

data\_message **=** data\_sort**.**head**(**10**)** # 提取疫情严重程度排名前十地区的信息

n **=** i**[:-**5**]** # 只提取国家名，不要后缀（.xlsx）

checknan**(**data\_message**[**'confirm'**])**

checknan**(**data\_message**[**'heal'**])**

checknan**(**data\_message**[**'dead'**])**

y1\_confirm **=** data\_message**[**'confirm'**]**

y1\_confirm **=** list**(**y1\_confirm**)**

y1\_confirm **=** **[**int**(**i**)** **for** i **in** y1\_confirm**]**

y2\_dead **=** data\_message**[**'dead'**]**

y2\_dead **=** list**(**y2\_dead**)**

y2\_dead **=** **[**int**(**i**)** **for** i **in** y2\_dead**]**

y3\_heal **=** data\_message**[**'heal'**]**

y3\_heal **=** list**(**y3\_heal**)**

y3\_heal **=** **[**int**(**i**)** **for** i **in** y3\_heal**]**

name\_list **=** **[]**

**for** j **in** data\_message**[**'name'**]:**

name\_list**.**append**(**j**)**

x **=** name\_list

color\_series **=** randomcolor**(**len**(**x**))**

page **=** Page**()** # 创建一个分页用于放所有的图像在这一分页上

* 1. **中国疫情地图可视化**
     1. **问题分析**

为了更直观地展示出中国的疫情情况，更直观地展示中国当前确诊人数、死亡人数等情况，我们针对部分爬取到的数据可视化，采用了map地图库，对生成的图像进行了分析，并新建了文件夹cn\_map\_view用于存放所生成的所有图像文件。

根据对数据的统计、分类和分析，我们绘制出了“中国各省今日新增疫情情况”、“中国累计确诊人数线形图”、“中国疫情地图（累计确诊人数）”、“中国疫情地图（累计死亡人数）”、“中国疫情地图（累计治愈人数）”、“中国疫情地图（现存确诊人数）”、“中国疫情地图（新增确诊人数）”的分布图。对这些图像进行分析，获取有用于我们的关于疫情各方面情况的信息，这样有利于我们了解疫情的发展趋势，对目前疫情形势进行分析，也有利于对未来疫情发展趋势进行预估，并采取相关有用的方案进行防控。

* + 1. **可视化分析**



图3-4-1 中国疫情地图可视化实现流程图

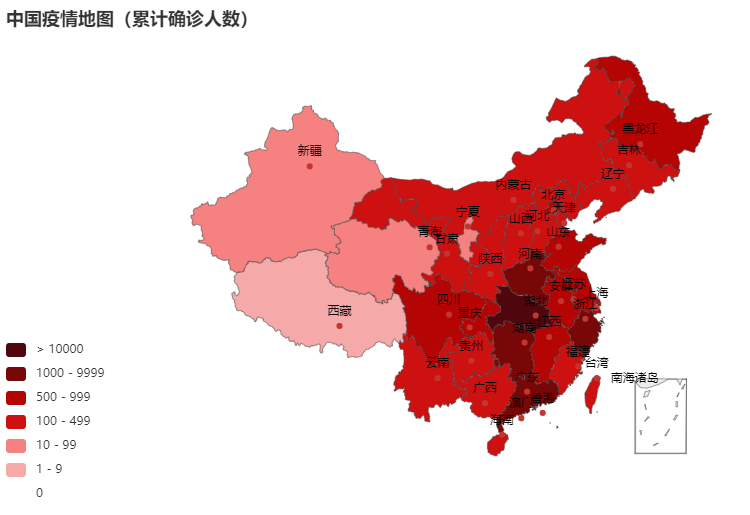
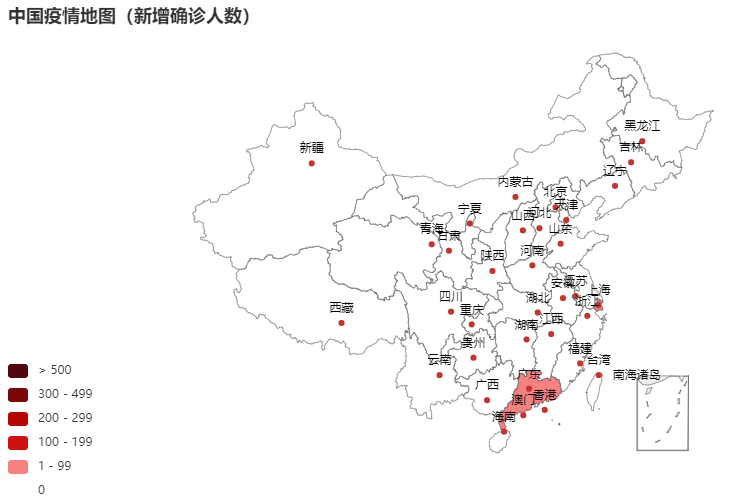


图3-4-2 中国疫情地图（累计确诊人数）

根据图3-4-2可以看出广东、湖北、湖南、河南以及浙江是中国疫情的重灾区，反观新疆、西藏、青海和宁夏感染情况相对其他省份较轻，之所以会产生差异，是因为湖北是此次疫情的爆发源头，湖北周围的省份会收到牵连，而西藏等地因海拔过高，地广人稀等环境因素的影响，这一优势阻挡了疫情的大规模扩散，患病人数也会相对较少。

图3-4-3 中国疫情地图（新增确诊人数）

从图3-4-3可以看出，今日仅有广东省和伤害新增确诊病例，主要是因为广东省是临海省份，是对外贸易的重要港口，因此每日的人流量巨大，在如此巨大的人流量下进行疫情防控是非常困难的。

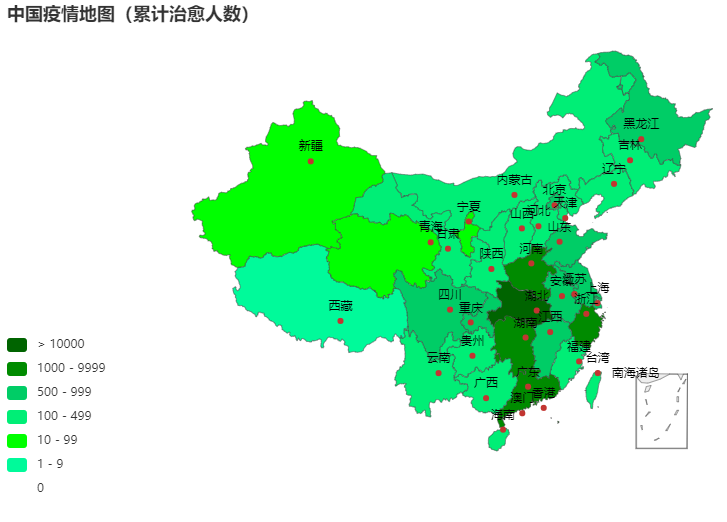


图3-4-4 中国疫情地图（累计治愈人数）

从图3-4-4可以看出，湖北省及其周围部分省份的治愈人数相对其他省份多许多，主要是因为国家和人们的重视，以及医护人员异地支援和防护用具的补缺，越来越多的病人得到有效及时的救治，而湖北救治工作更是重中之重。

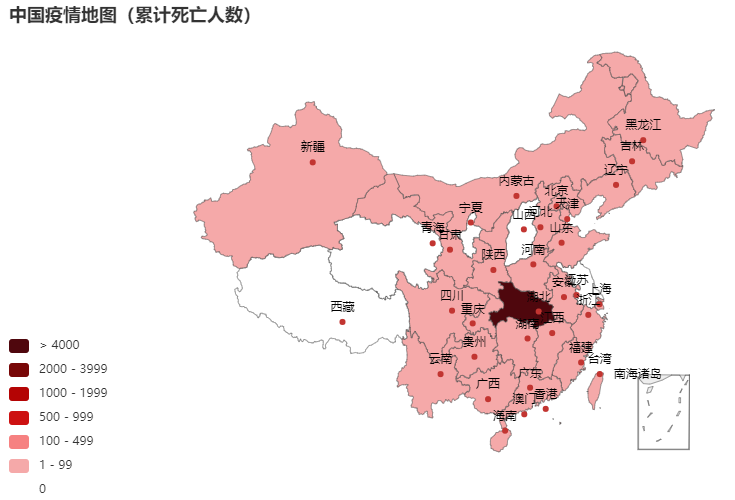


图3-4-5 中国疫情地图（累计死亡人数）

从图3-4-5可以看出，湖北省的死亡人数远超其他省份，主要是因为疫情爆发的前期，由于新型冠状病毒来的突然又迅猛，对于病毒的了解不足，加上当时人们的防范意识不强，对疫情形势的严峻程度不了解，以及医疗防护用具的供应不足和医疗人员的缺乏，让新冠肺炎患者错失了及时确诊并及时治疗的最佳时机，导致死亡人数增多。

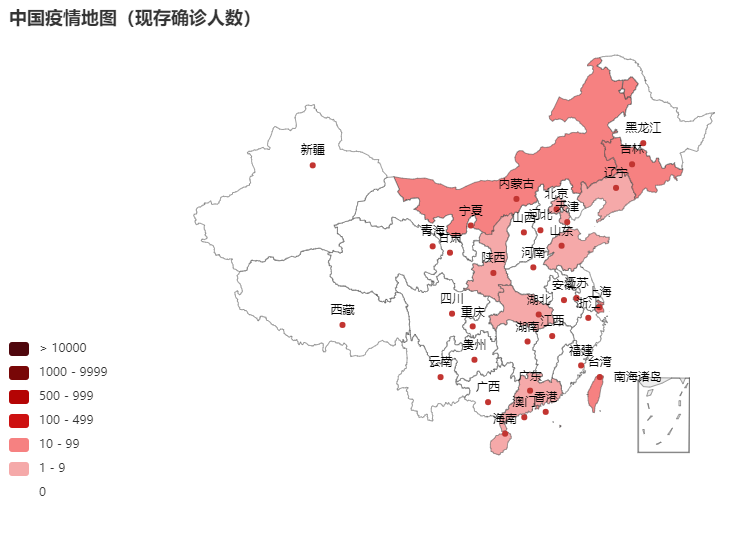


图3-4-6 中国疫情地图（现存确诊人数）

从图3-4-6可以看出，随着国家和人民的重视，对疫情防控工作的加强，以及医护人员异地支援和防护用具的补缺，许多感染病人得到有效及时的救治，现存确诊人数大大减少，这是人们和国家共同努力的成果。

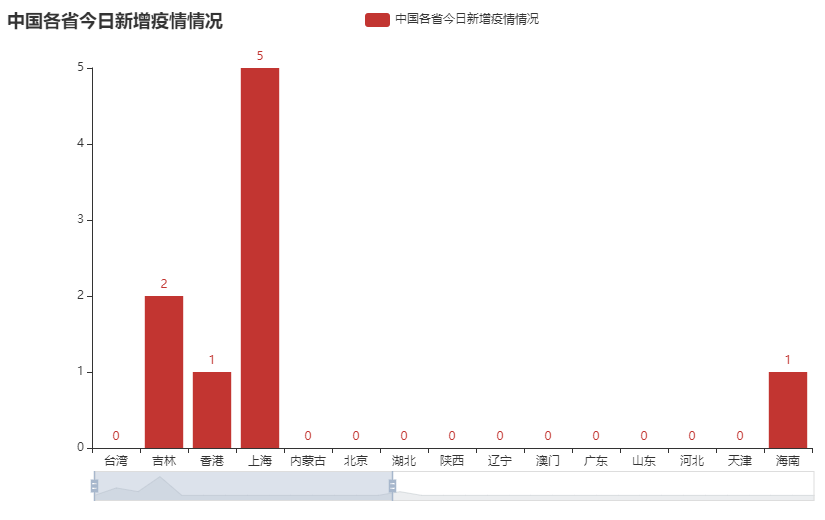
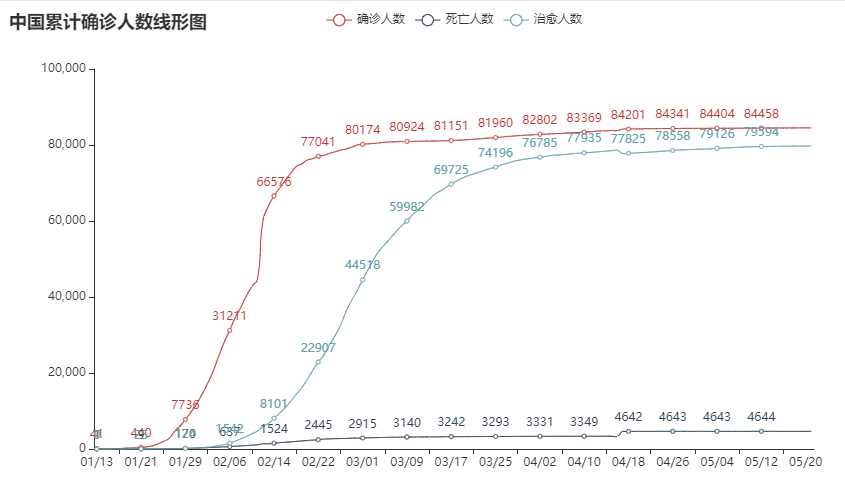
****

图3-4-7 中国各省今日新增疫情情况

图3-4-8 中国累计确诊人数线性图

从图3-4-8中可看出从1月21日开始至2月22日，中国确诊人数迸发式增长，而从2月22日至今，中国累计确诊人数和治愈人数趋于平缓，2月22日前后增长差异巨大，可能是因为检测新冠肺炎的方式发生了变化以及对新冠肺炎确诊的排查范围进行了扩大。

* + 1. **实现过程**

采用pandas.read\_excel()，读取excel表里已收集到的数据，再通过pandas.DataFrame()将其保存为表单形式。

代码 3-4-1

**import** pyecharts**.**options **as** opts

**from** pyecharts**.**charts **import** Line**,** Bar**,** Map

**import** pandas **as** pd

**def** \_\_init\_\_**(**self**):**

self**.**data\_pro\_tota **=** pd**.**DataFrame**(**pd**.**read\_excel**(**'data\中国各省市(区)总体疫情信息\中国各省市(区)总体疫情信息.xlsx'**))**

self**.**data\_pro\_newly **=** pd**.**DataFrame**(**pd**.**read\_excel**(**'data\中国各省市(区)总体疫情信息\中国各省市(区)今日新增疫情信息.xlsx'**))**

self**.**data\_tota **=** pd**.**DataFrame**(**pd**.**read\_excel**(**'data\中国总体历史疫情信息\历史总体信息.xlsx'**))**

**中国疫情地图（累计确诊人数）：**

代码4-4-2

**def** china\_map\_confirm**(**self**):**

map\_confirm **=** **(**

Map**()**

**.**add**(**""**,** **[**list**(**z**)** **for** z **in** zip**(**self**.**data\_pro\_tota**.**name**,** self**.**data\_pro\_tota**.**confirm**)],** "china"**)**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**"中国疫情地图（累计确诊人数）"**),**

visualmap\_opts **=** opts**.**VisualMapOpts**(**is\_piecewise **=** **True,** pieces**=[**

**{**"min"**:**10000**,** "color" **:** "#4F070D"**},**

**{**"min"**:** 1000**,**"max"**:**9999**,** "color"**:** "#780707"**},**

**{**"min"**:** 500**,** "max"**:** 999**,** "color"**:** "#B40404"**},**

**{**"min"**:** 100**,** "max"**:** 499**,** "color"**:** "#CD1111"**},**

**{**"min"**:** 10**,** "max"**:** 99**,**"color"**:** "#F68181"**},**

**{**"min"**:**1**,**"max"**:** 9**,** "color"**:** "#F5A9A9"**},**

**{**"max"**:** 0**,** "min"**:** 0**,** "label"**:** "0"**,** "color"**:** "#FFFFFF"**},**

**]))**

**)**

**return** map\_confirm

**中国各省今日新增疫情情况：**

代码4-4-3

**def** bar\_now**(**self**):**

bar **=** **(**

Bar**()**

**.**add\_xaxis**(**list**(**self**.**data\_pro\_newly**.**name**))**

**.**add\_yaxis**(**"中国各省今日新增疫情情况"**,** list**(**self**.**data\_pro\_newly**.**confirm**))**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**"中国各省今日新增疫情情况"**),**

datazoom\_opts**=**opts**.**DataZoomOpts**(),**

**)**

**)**

grid\_chart **=** Grid**()**

grid\_chart**.**add**(**

bar**,**

grid\_opts**=**opts**.**GridOpts**(**

pos\_left**=**"3%"**,** pos\_right**=**"1%"**,** height**=**"20%"

**),**

**)**

**return** bar

**调用各函数绘制图像并保存：**

代码4-4-4

**def** get\_china\_map\_confirm**(**self**):**

m **=** self**.**china\_map\_confirm**()**

**return** m**.**render**(**"cn\_map\_view\中国疫情地图（累计确诊人数）.html"**)**

**def** get\_china\_map\_NewAddConfirm**(**self**):**

m **=** self**.**china\_map\_NewAddConfirm**()**

**return** m**.**render**(**"cn\_map\_view\中国疫情地图（新增确诊人数）.html"**)**

**def** get\_china\_map\_nowConfirm**(**self**):**

m **=** self**.**china\_map\_nowConfirm**()**

**return** m**.**render**(**"cn\_map\_view\中国疫情地图（现存确诊人数）.html"**)**

**def** get\_china\_map\_heal**(**self**):**

m **=** self**.**china\_map\_heal**()**

**return** m**.**render**(**"cn\_map\_view\中国疫情地图（累计治愈人数）.html"**)**

**def** get\_china\_map\_dead**(**self**):**

m **=** self**.**china\_map\_dead**()**

**return** m**.**render**(**"cn\_map\_view\中国疫情地图（累计死亡人数）.html"**)**

**def** get\_bar\_now**(**self**):**

b **=** self**.**bar\_now**()**

**return** b**.**render**(**"cn\_map\_view\中国各省今日新增疫情情况.html"**)**

**def** get\_china\_line**(**self**):**

l **=** self**.**china\_line**()**

**return** l**.**render**(**"cn\_map\_view\中国累计确诊人数线形图.html"**)**

**def** main**(**self**):**

self**.**get\_china\_map\_confirm**()**

self**.**get\_china\_map\_nowConfirm**()**

self**.**get\_china\_map\_NewAddConfirm**()**

self**.**get\_china\_map\_heal**()**

self**.**get\_china\_map\_dead**()**

self**.**get\_bar\_now**()**

self**.**get\_china\_line**()**

**if** \_\_name\_\_ **==** '\_\_main\_\_'**:**

map **=** View**()**

map**.**main**()**

* 1. **世界疫情及地级市疫情地图可视化**
     1. **问题分析**

通过绘制世界疫情以及地级市疫情一系列交互地图，以这种直观的方式，帮助我们更好的了解疫情发展变化，通过最直观清楚的确诊数据、疑似数据、治愈数据和死亡数据统计总体上了解疫情的分布趋势，在中国疫情分布地图上，以省级行政区为单位，我们可以发现，疫情爆发地湖北省疫情最为严重，其他省份紧随其后，且人口密度与确诊病例人数之间存在明显相关性，人口密度越大，疫情越容易传播，防控工作难度越高。我们对爬取到的数据进行可视化，使用了pyecharts库的Bar, Line, Map, Pie生成以图像并保存为.html的形式更为直观的表现的人们的眼中。

根据所获得的数据文件，绘制出了疫情信息地图和国内疫情走向折线图等图像，疫情信息地图分为世界、省级和县市级三级呈现。这有助于我们更容易看懂和理解数据，以及传达的信息更明确，充分了解疫情发展情况。

* + 1. **时间轴图像分析**

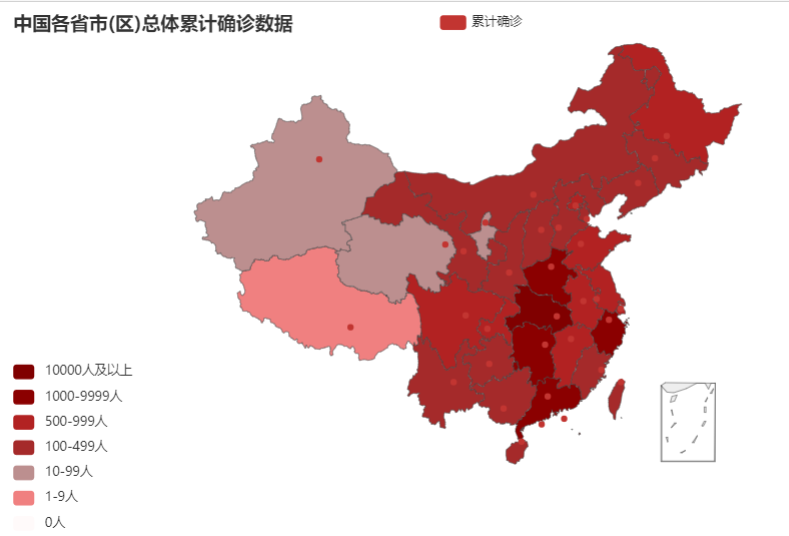


图3-5-1 中国各省市（区）总体累计确诊数据

根据颜色的深度来区分每个省份累计确诊人数的多少，从上图可以很明显的看出疫情由湖北为中心向四周蔓延，疫情由输入性向扩散性过渡，波及的范围越来越广泛，广东作为人口流量大省，外来输入病例引起聚集性感染，疫情严重程度不输发源地武汉，此图可以很直观的看出从疫情开始我国各省市（区）的累计确诊情况和各省份疫情严重程度。



图3-5-2 中国各省市（区）今日新增疫情信息

经过严防死守的防控措施，让疫情防止大范围的扩散。督促湖北省和武汉市依法采取最严格的防控措施，加强农贸市场监管和野生动物管控，内防扩散、外防输出，劝导公众宜散不宜聚，最大程度减少公众聚集性活动，避免聚集性疫情发生，坚决落实疫情防控措施。织密织牢不明原因肺炎监测、发热门诊预检分诊两张网，做好病例救治各项准备，及时发现并有效处置疫情。加大环境卫生整治力度，结合春节期间人员流动性加大的特点，因地制宜落实车站、机场、码头等重要场所和汽车、火车、飞机等密闭交通工具的通风、消毒、测体温等措施。数据更新至5月23号除去吉林省新增确诊病例2例和四川省新增确诊病例2例我国已基本没有新增确诊人员，疫情防控基本稳定。

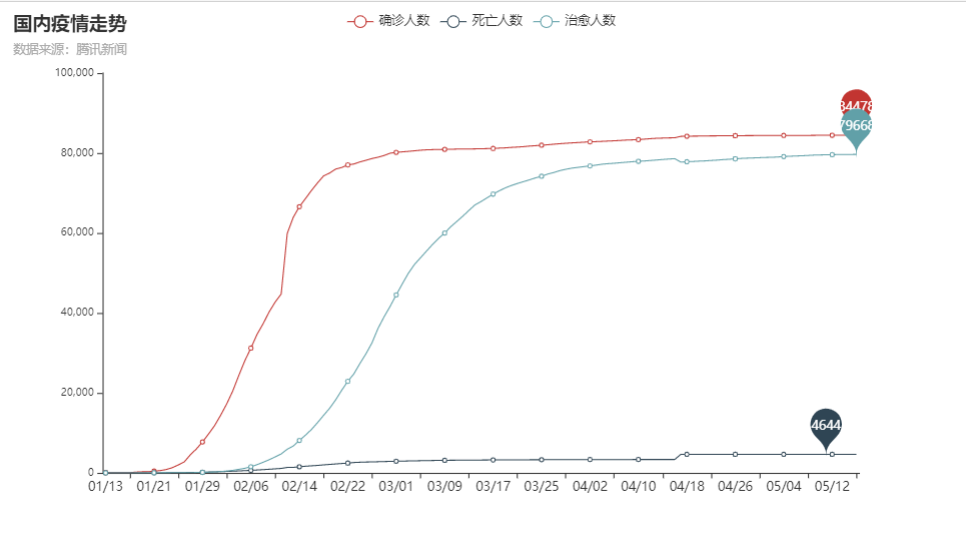


图3-5-3 中国总体历史疫情信息

从图3-5-3可以看出一月份开始出现感染者，开始并不知情是新型冠状病毒引起的，人们的防范意识不强，对疫情形势的严峻程度不了解，没有得到有效的控制和防护，二月份疫情开始大范围爆发感染人数持续增加，随着时间的增长，疫情防控的不断加强，新增感染人数逐渐减少，治愈的人数也不断增加，对疫情的防控趋于稳定。

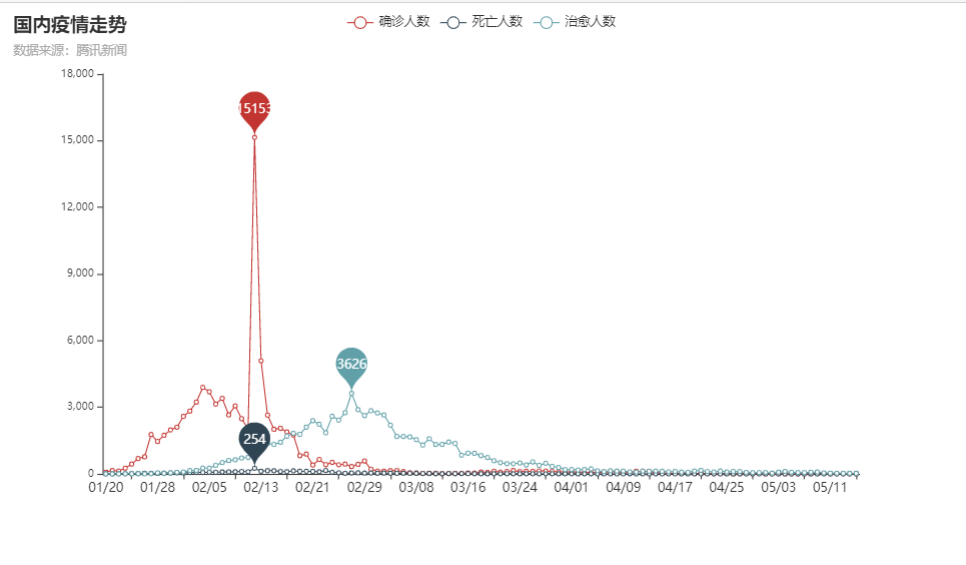


图3-5-4 国内疫情历史每日新增信息

观察图3-5-4可得到，二月份疫情大范围爆发在2月13日新增确诊人数达历史最高，随着疫情防控的不断加强，每日新增确诊人数大幅度减少，至5月份已基本没有新增病例，大部分为境外输入病例。

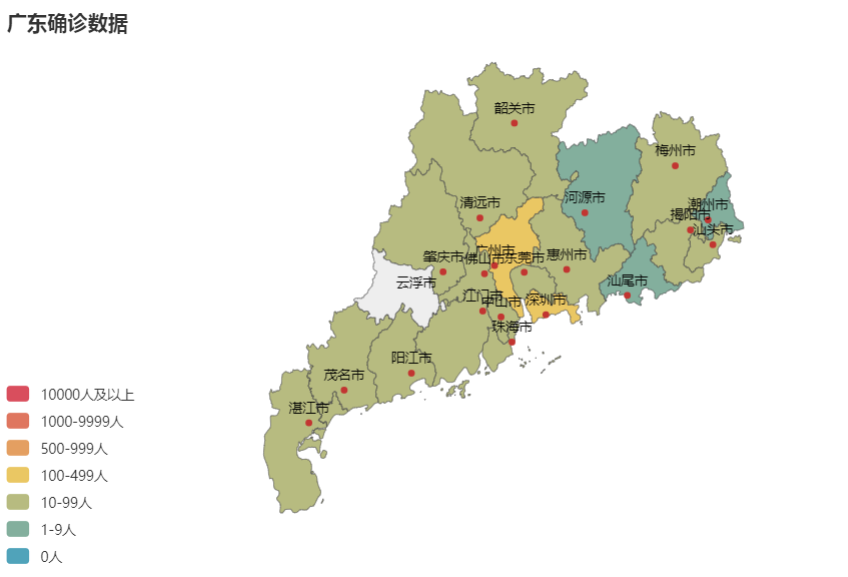


图3-5-5 中国各省市(区)疫情信息（广东、湖北）

从图3-5-5中可看出，根据颜色的深度来区分每个市区累计确诊人数的多少，广东省内疫情较严重地区为广州市和深圳市，河源市、潮州市、汕尾市确诊人数较少，广州和深圳为广东的经济发达地区，人口流动量大，导致疫情扩散较为严重。

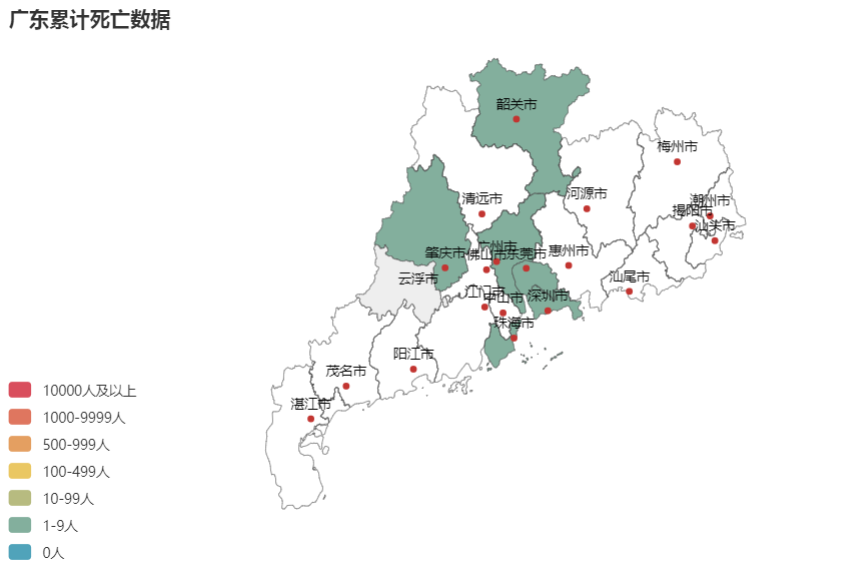


图3-5-6 广东累计死亡数据

结合图3-5-5、图3-5-6可看出，广东省内新冠肺炎患者已治愈的占比很高，死亡数占比较低，省内累积病例已基本痊愈。疫情得到有效控制，得益于政府劝导公众宜散不宜聚，最大程度减少公众聚集性活动，避免聚集性疫情发生，坚决落实疫情防控措施，以及人们的积极配合和防范意识、卫生健康安全意识增强。

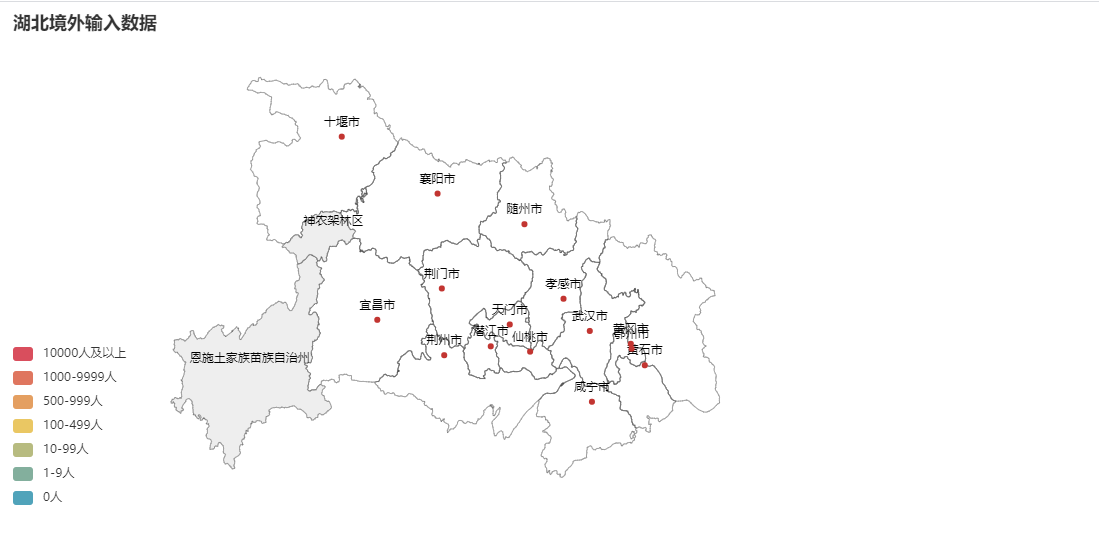


图 3-5-7湖北省当日境外输入数据图

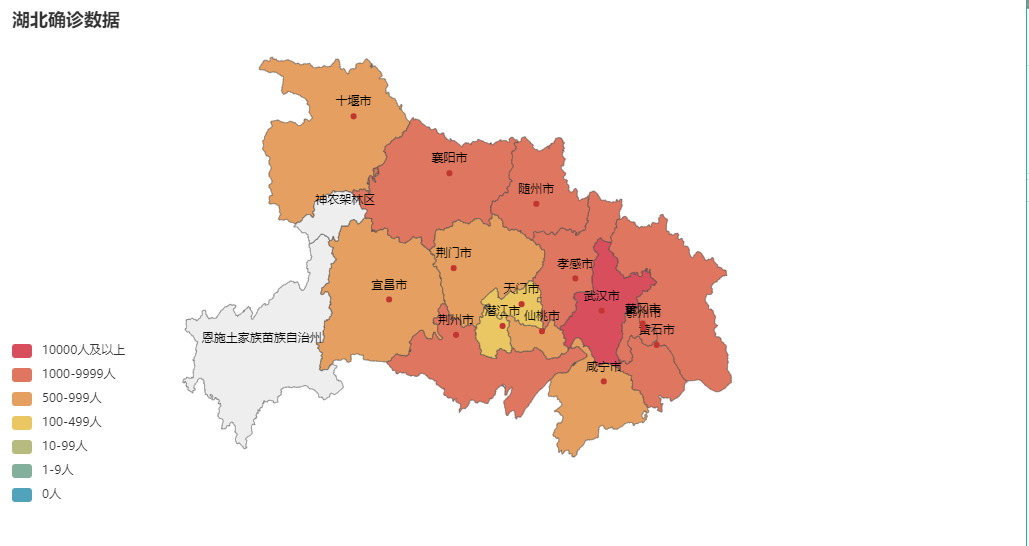


图3-5-8 湖北省各地区的累计确诊人数分布图

从图 3-5-7中可以看出湖北省内当日没有境外病例输入的情况，这得益于国家实行一个外防输入，内防扩散，严防输出的总原则。

从图3-5-8中可以看出湖北省除了恩施土家族苗族自治州和神农架林区外，其他的地区累计确诊人数都在100以上，天门市和潜江市比较轻，人数在100左右，严重的如襄阳市、随州市、孝感市等几个市区累计确诊人数都超过了1000人，最严重的武汉市超过了10000人，达到了50340人，这是因为疫情初期爆发的时候恰巧在春节前，很多人回家过年，而且武汉人口流量大，人口基数大。

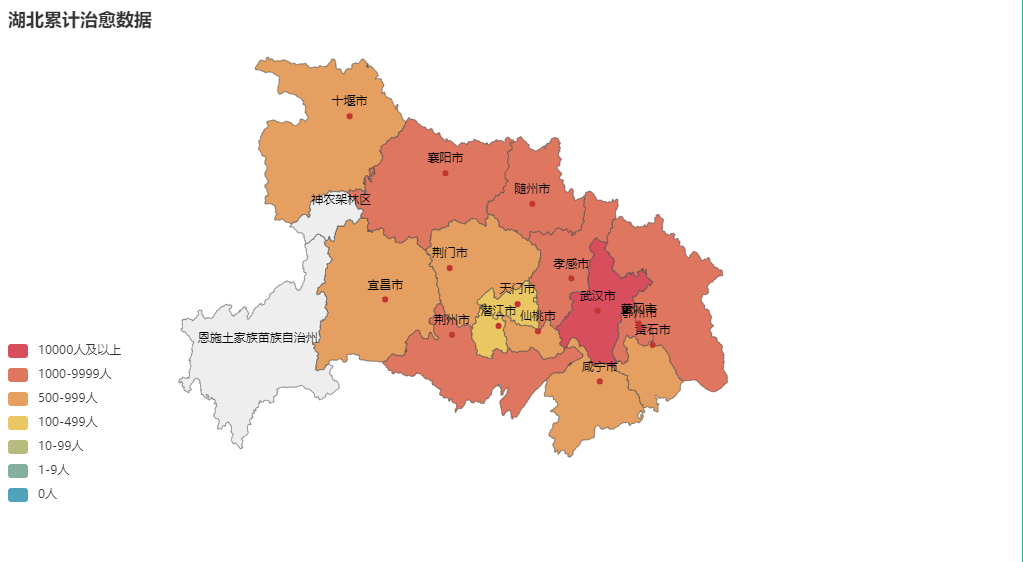


图3-5-9 湖北省各地区累计治愈人数图

因为恩施土家族苗族自治州和神农架林区没有发生疫情，所以治愈人数为0，潜江市和天门市的累计治愈人数分别是198和496人，疫情最严重的武汉市累计治愈人数为46464人。这得益于国家和全国人民的努力，一方有难，八方支援。国家财政大力支援，还派遣大量军医赶赴一线，全国各个省市区都有医疗队奔赴灾区，去救治患者。

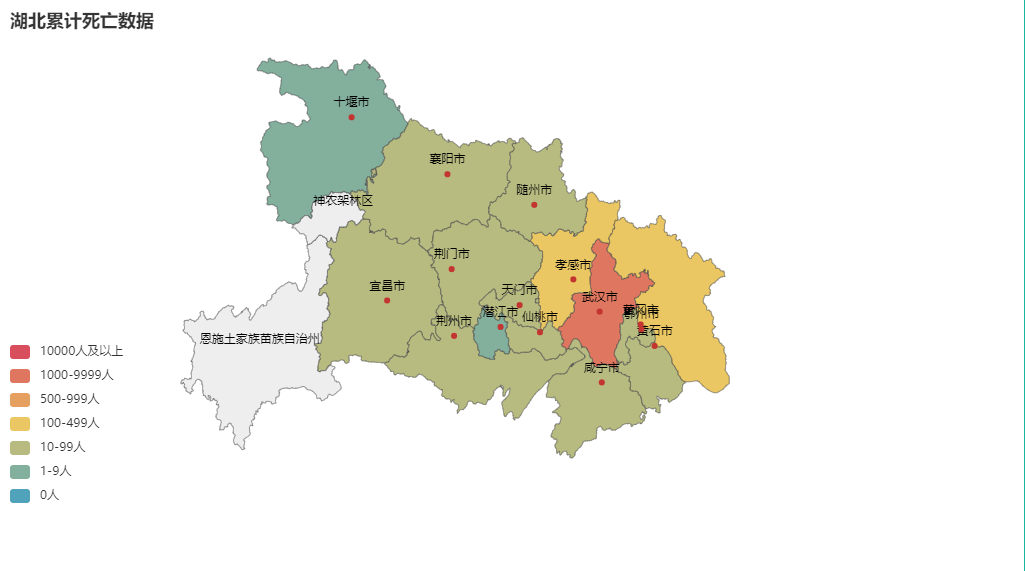


图 3-5-10 湖北省累计死亡数据分布图

从图中可以看出恩施土家族苗族自治州和神农架林区没有死亡人数，潜江市和十堰市的累计死亡人数在10以内，襄阳市、宜昌市、荆门市等十个市区的累计死亡人数在10-99之间，孝感市和黄冈市的累计死亡人数超过了100，分别是129人和125人，而疫情最严重的的武汉市则达到了3869人。这是因为湖北地区最早爆发疫情，其他地区有预防，所以其他地区比湖北地区的死亡人数要比较少。

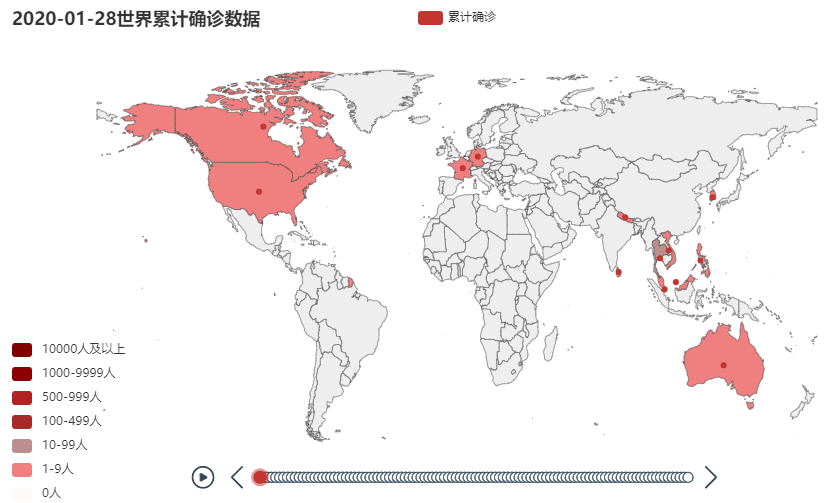


图 3-5-11 世界累计确诊数据

根据爬取的数据绘制了随时间的推移世界累计确诊情况图。从这个动态图中，可以看出，随着时间变化，疫情是以中国为中心向周围扩散，逐渐扩散至全球，几乎无一幸免；随着时间的推移，各国累计确诊人数也越来越多，疫情正以一种无法阻挡的趋势侵略全球。

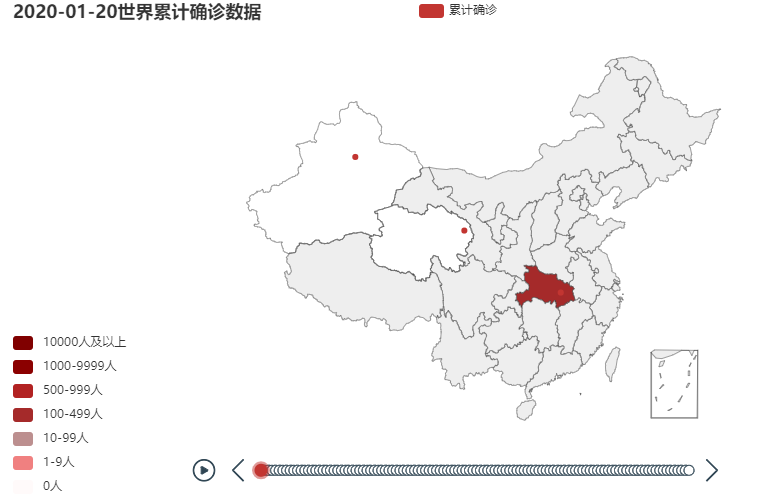


图3-5-12 中国累计确诊数据

根据爬取的数据绘制了随时间的推移中国累计确诊情况图。从这个动态图中，可以看出，随着时间变化，疫情是以武汉为中心向周围扩散，逐渐扩散至全国；随着时间的推移，各省累计确诊人数也越来越多，疫情正以一种无法阻挡的趋势侵略全国。

* + 1. **实现过程**

**绘制世界地图：**

代码3-5-1

chinaMap **=** **(** # 进行画图

Map**()** **.**

**.**add**(**"累计确诊"**,** **[**list**(**z**)** **for** z **in** zip**(**data\_name**,** data\_confirm**)],** maptype**=**"world"**)** #添加并遍历数据文件的data\_name和data\_confirm,地图类型为世界地图

**.**set\_series\_opts**(**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**set\_global\_opts**(**

title\_opts**=**opts**.**TitleOpts**(**title**=**"世界累计确诊数据"**),**

visualmap\_opts**=**opts**.**VisualMapOpts**(**is\_piecewise**=True,** #设置颜色分段显示

pieces**=[{**"max"**:** 0**,** "label"**:** '0人'**,** "color"**:** "#FFFAFA"**},**

**{**"min"**:** 1**,** "max"**:** 9**,** "label"**:** '1-9人'**,** "color"**:** "#F08080"**},**

**{**"min"**:** 10**,** "max"**:** 99**,** "label"**:** '10-99人'**,** "color"**:** "#BC8F8F"**},**

**{**"min"**:** 100**,** "max"**:** 499**,** "label"**:** '100-499人'**,** "color"**:** "#A52A2A"**},**

**{**"min"**:** 500**,** "max"**:** 999**,** "label"**:** '500-999人'**,** "color"**:** "#B22222"**},**

**{**"min"**:** 1000**,** "max"**:** 9999**,** "label"**:** '1000-9999人'**,** "color"**:** "#8B0000"**},**

**{**"min"**:** 10000**,** "label"**:** '10000人及以上'**,** "color"**:** "#800000"**}]),**

**)**

**)**

self**.**path **=** self**.**file **+** '\\photos\\世界总体疫情信息'

isExists **=** os**.**path**.**exists**(**self**.**path**)** # 判断当前目录是否有世界总体疫情信息文件夹，如果没有则创建

**if** **not** isExists**:**

os**.**mkdir**(**self**.**path**)** # 在当前路径创建新文件夹analisis,用于存放生成的图像数据

chinaMap**.**render**(**'photos/世界总体疫情信息/世界总体疫情信息.html'**)** # 保存为html格式

**中国各省市(区)总体疫情信息：**

代码3-5-2 读取数据并处理数据

data **=** pd**.**read\_excel**(**'data/中国各省市(区)总体疫情信息/中国各省市(区)总体

信息**.**xlsx', index\_col=False) #读取数据

data\_name **=** data**[**'name'**]** #获取data\_name数据

data\_confirm **=** data**[**'confirm'**]** #获取data\_confirm数据

**for** z **in** zip**(**data\_name**,** data\_confirm**):** #将这两个类型的数据整合成一个二维数组

**print(**list**(**z**))**

代码3-5-3 绘制图像

chinaMap **=** **(** #绘制图像

Map**()**

**.**add**(**'累计确诊'**,** **[**list**(**z**)** **for** z **in** zip**(**data\_name**,** data\_confirm**)],** maptype**=**'china'**)** #遍历数据文件名为data\_name和data\_confirm的数据，地图类型为中国地图

**.**set\_series\_opts**(**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**#给每个省份标点

**.**set\_global\_opts**(**

title\_opts**=**opts**.**TitleOpts**(**title**=**"中国各省市(区)总体累计确诊数据"**),** #为该图像取标题

visualmap\_opts**=**opts**.**VisualMapOpts**(**is\_piecewise**=True,** #采用视觉印射配置，并将颜色分段显示

pieces**=[{**"max"**:** 0**,** "label"**:** '0人'**,** "color"**:** "#FFFAFA"**},**

**{**"min"**:** 1**,** "max"**:** 9**,** "label"**:** '1-9人'**,** "color"**:** "#F08080"**},**

**{**"min"**:** 10**,** "max"**:** 99**,** "label"**:** '10-99人'**,** "color"**:** "#BC8F8F"**},**

**{**"min"**:** 100**,** "max"**:** 499**,** "label"**:** '100-499人'**,** "color"**:** "#A52A2A"**},**

**{**"min"**:** 500**,** "max"**:** 999**,** "label"**:** '500-999人'**,** "color"**:** "#B22222"**},**

**{**"min"**:** 1000**,** "max"**:** 9999**,** "label"**:** '1000-9999人'**,** "color"**:** "#8B0000"**},**

**{**"min"**:** 10000**,** "label"**:** '10000人及以上'**,** "color"**:** "#800000"**}]),**

**)**

**)**

代码3-5-4 保存为html文件

chinaMap**.**render**(**'photos/中国各省市(区)总体疫情信息/中国各省市(区)总体疫情

息**.**html') #将图像保存为html文件

**中国总体历史疫情信息：**

代码3-5-5 生成日期列表并将列表对x进行赋值

**for** j **in** data**[**'date'**]:** #生成日期列表

date\_list**.**append**(**j**)**

x **=** date\_list #生成的日期列表命名为x

代码3-5-6 绘制2D的折线图

lines **=** **(** #绘制2D的折线图

Line**()**

**.**add\_xaxis**(**xaxis\_data**=**x**)** #将日期列表作为X轴

**.**add\_yaxis**(**series\_name**=**'确诊人数'**,** y\_axis**=**y\_confirm**,** is\_symbol\_show**=True,** #分别将确诊人数、确诊人数、确诊人数作为Y轴

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**add\_yaxis**(**series\_name**=**'确诊人数'**,** y\_axis**=**y\_dead**,** is\_symbol\_show**=True,**

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**add\_yaxis**(**series\_name**=**'确诊人数'**,** y\_axis**=**y\_heal**,** is\_symbol\_show**=True,**

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**"国内疫情走势"**,** subtitle**=**"数据来源：腾讯新闻"**),** #命名标题

yaxis\_opts**=**opts**.**AxisOpts**(**axislabel\_opts**=**opts**.**LabelOpts**(**font\_size**=**10**,** interval**=**3**)))** #设置x轴标签全部显示

**)**

**各国历史疫情信息：**

代码3-5-7

**import** os 7

**import** pandas **as** pd

**import** numpy **as** np

**import** pyecharts**.**options **as** opts

**from** pyecharts**.**charts **import** Bar**,** Line**,** Map**,** Pie

path **=** 'C:\\Users\\diaosi\\Desktop\\Python 爬虫助力疫情数据追踪项目代码\\Python 爬虫助力疫情数据追踪项目代码\\data\\各国历史疫情信息'

Filelist **=** **[]** #遍历path路径下所有文件夹并将文件名合并在一起

**for** home**,** dirs**,** files **in** os**.**walk**(**path**):**

**for** filename **in** files**:**

Filelist**.**append**(**filename**)**

**print(**Filelist**)**

**for** i **in** Filelist**:**

data **=** pd**.**read\_excel**(**'data/各国历史疫情信息/%s' **%** **(**i**),** index**=False)** #读取文件

n **=** i**[:-**5**]**

y1\_confirm **=** data**[**'confirm'**]** #获取y1数据

y2\_heal **=** data**[**'heal'**]** #获取y2数据

y3\_dead **=** data**[**'dead'**]** #获取y3数据

date\_list **=** **[]**

**for** j **in** data**[**'date'**]:**

date\_list**.**append**(**j**)** #获取x轴数据

x **=** date\_list

lines **=** **(** #绘制折线图

Line**()**

**.**add\_xaxis**(**xaxis\_data**=**x**)** #xaxis\_data为x轴，y\_axis为y轴

**.**add\_yaxis**(**series\_name**=**'确诊人数'**,** y\_axis**=**y1\_confirm**,** is\_symbol\_show**=True,** #不显示数据

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**add\_yaxis**(**series\_name**=**'治愈人数'**,** y\_axis**=**y2\_heal**,** is\_symbol\_show**=True,**

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**add\_yaxis**(**series\_name**=**'死亡人数'**,** y\_axis**=**y3\_dead**,** is\_symbol\_show**=True,**

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**"%s"**%** **(**n**)+**"疫情走势"**,** subtitle**=**"数据来源：腾讯新闻"**),**

yaxis\_opts**=**opts**.**AxisOpts**(**axislabel\_opts**=**opts**.**LabelOpts**(**font\_size**=**10**,** interval**=**3**)))**

**)**

lines**.**render\_notebook**()**

lines**.**render**(**'photos/各国历史疫情信息/%s.html' **%** **(**n**),** index**=False)** #保存为html文件

**中国各省的城市历史疫情信息：**

代码3-5-8

lines **=** **(**

Line**()**

**.**add\_xaxis**(**xaxis\_data**=**x**)** #添加以时间为X轴

**.**add\_yaxis**(**series\_name**=**'确诊人数'**,** y\_axis**=**y\_confirm**,** is\_symbol\_show**=True,** #添加以确诊人数、死亡人数、治愈人数作为Y轴

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**add\_yaxis**(**series\_name**=**'死亡人数'**,** y\_axis**=**y\_dead**,** is\_symbol\_show**=True,**

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**add\_yaxis**(**series\_name**=**'治愈人数'**,** y\_axis**=**y\_heal**,** is\_symbol\_show**=True,**

label\_opts**=**opts**.**LabelOpts**(**is\_show**=False),**

markpoint\_opts**=**opts**.**MarkPointOpts**(**data**=[**opts**.**MarkPointItem**(**type\_**=**"max"**),** **]))**

**.**set\_global\_opts**(**title\_opts**=**opts**.**TitleOpts**(**title**=**"%s"**%** **(**n**)+**"%s" **%** **(**z**)+**"疫情走势"**,** subtitle**=**"数据来源：腾讯新闻"**),**# 全局设置，添加标题title和子标题subtitle

yaxis\_opts**=**opts**.**AxisOpts**(**axislabel\_opts**=**opts**.**LabelOpts**(**font\_size**=**10**,** interval**=**3**)))** #设置x轴标签全部显示

**)**

**拆分时间轴：**

代码3-5-9

**import** pandas **as** pd

data **=** pd**.**read\_excel**(**'photos/世界各国合成表.xlsx'**,** index\_col**=False)**#读取世界各国合成表

data**.**columns

data**[**'date'**]** **=** pd**.**PeriodIndex**(**data**[**'date'**],** freq**=**'D'**)**

class\_list **=** list**(**data**[**'date'**].**drop\_duplicates**())**# 拆分数据，并写出

**for** i **in** class\_list**:**

data1 **=** data**[**data**[**'date'**]** **==** i**]**

data1**.**to\_excel**(**'photos/世界各国时间轴/%s.xlsx' **%** **(**i**),** index**=False)**

data **=** pd**.**read\_excel**(**'photos/中国各省市(区)合成表.xlsx'**,** index\_col**=False)**#读取中国各省市（区）合成表

data**.**columns

data**[**'date'**]** **=** pd**.**PeriodIndex**(**data**[**'date'**],** freq**=**'D'**)**

class\_list **=** list**(**data**[**'date'**].**drop\_duplicates**())**# 拆分数据，并写出

**for** i **in** class\_list**:**

data1 **=** data**[**data**[**'date'**]** **==** i**]**

data1**.**to\_excel**(**'photos/中国时间轴/%s.xlsx' **%** **(**i**),** index**=False)**

**时间轴图像：**

代码3-5-10

**import** pandas **as** p

**import** pyecharts**.**options **as** opts

**from** pyecharts**.**charts **import** Map**,** Timeline

data **=** pd**.**read\_excel**(**'photos/世界各国合成表.xlsx'**,** index\_col**=False)**#读取文件

data**.**columns#查看列名

data**[**'date'**]** **=** pd**.**PeriodIndex**(**data**[**'date'**],** freq**=**'D'**)**#对时间日期格式处理和转换

class\_list **=** list**(**data**[**'date'**].**drop\_duplicates**())**#转换为列表数据类型并去重

#创建 Timeline 对象

t1 **=** Timeline**()**

#获取世界各国国家名称和累计确诊人数

**for** i **in** class\_list**:**

data1 **=** pd**.**read\_excel**(**'photos/世界各国时间轴/%s.xlsx' **%** **(**i**),** index**=False)**#读取世界各国时间轴文件夹下的所有xlsx表

data\_name **=** data1**[**'English\_name'**]**

data\_confirm **=** data1**[**'confirm'**]**

#画图

chinaMap **=** **(**

Map**()**

**.**add**(**"累计确诊"**,** **[**list**(**z**)** **for** z **in** zip**(**data\_name**,** data\_confirm**)],** maptype**=**"world"**)**

**.**set\_series\_opts**(**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**#不显示具体的数据值

**.**set\_global\_opts**(**

title\_opts**=**opts**.**TitleOpts**(**title**=**"{}世界累计确诊数据"**.**format**(**i**)),**

#视觉映射配置项

visualmap\_opts**=**opts**.**VisualMapOpts**(**

# 是否为分段型

is\_piecewise**=True,**

# 自定义的每一段数据的范围，以及每一段的文字，以及每一段的特别的样式。

pieces**=[{**"max"**:** 0**,** "label"**:** '0人'**,** "color"**:** "#FFFAFA"**},**

**{**"min"**:** 1**,** "max"**:** 9**,** "label"**:** '1-9人'**,** "color"**:** "#F08080"**},**

**{**"min"**:** 10**,** "max"**:** 99**,** "label"**:** '10-99人'**,** "color"**:** "#BC8F8F"**},**

**{**"min"**:** 100**,** "max"**:** 499**,** "label"**:** '100-499人'**,** "color"**:** "#A52A2A"**},**

**{**"min"**:** 500**,** "max"**:** 999**,** "label"**:** '500-999人'**,** "color"**:** "#B22222"**},**

**{**"min"**:** 1000**,** "max"**:** 9999**,** "label"**:** '1000-9999人'**,** "color"**:** "#8B0000"**},**

**{**"min"**:** 10000**,** "label"**:** '10000人及以上'**,** "color"**:** "#800000"**}]),**

**)**

**)**

#用 Timeline 来画时间线

t1**.**add**(**chinaMap**,** ""**.**format**(**i**))**

t1**.**render**(**'photos/时间轴图像/世界总体疫情变化.html'**)**#保存文件

data **=** pd**.**read\_excel**(**'photos/中国各省市(区)合成表.xlsx'**,** index\_col**=False)**#读取文件

data**.**columns

data**[**'date'**]** **=** pd**.**PeriodIndex**(**data**[**'date'**],** freq**=**'D'**)**#对时间日期格式处理和转换

class\_list **=** list**(**data**[**'date'**].**drop\_duplicates**())**#转换为列表数据类型并去重

#创建 Timeline 对象

t1 **=** Timeline**()**

#获取世界各国国家名称和累计确诊人数

**for** i **in** class\_list**:**

data1 **=** pd**.**read\_excel**(**'photos/中国时间轴/%s.xlsx' **%** **(**i**),** index**=False)**#读取世界各国时间轴文件夹下的所有xlsx表

data\_name **=** data1**[**'country'**]**

data\_confirm **=** data1**[**'confirm'**]**

#画图

chinaMap **=** **(**

Map**()**

**.**add**(**"累计确诊"**,** **[**list**(**z**)** **for** z **in** zip**(**data\_name**,** data\_confirm**)],** maptype**=**"china"**)**

**.**set\_series\_opts**(**label\_opts**=**opts**.**LabelOpts**(**is\_show**=False))**

**.**set\_global\_opts**(**

title\_opts**=**opts**.**TitleOpts**(**title**=**"{}世界累计确诊数据"**.**format**(**i**)),**

#视觉映射配置项

visualmap\_opts**=**opts**.**VisualMapOpts**(**

# 是否为分段型

is\_piecewise**=True,**

# 自定义的每一段数据的范围，以及每一段的文字，以及每一段的特别的样式。

pieces**=[{**"max"**:** 0**,** "label"**:** '0人'**,** "color"**:** "#FFFAFA"**},**

**{**"min"**:** 1**,** "max"**:** 9**,** "label"**:** '1-9人'**,** "color"**:** "#F08080"**},**

**{**"min"**:** 10**,** "max"**:** 99**,** "label"**:** '10-99人'**,** "color"**:** "#BC8F8F"**},**

**{**"min"**:** 100**,** "max"**:** 499**,** "label"**:** '100-499人'**,** "color"**:** "#A52A2A"**},**

**{**"min"**:** 500**,** "max"**:** 999**,** "label"**:** '500-999人'**,** "color"**:** "#B22222"**},**

**{**"min"**:** 1000**,** "max"**:** 9999**,** "label"**:** '1000-9999人'**,** "color"**:** "#8B0000"**},**

**{**"min"**:** 10000**,** "label"**:** '10000人及以上'**,** "color"**:** "#800000"**}]),**

**)**

**)**

#用 Timeline 来画时间线

t1**.**add**(**chinaMap**,** ""**.**format**(**i**))**

t1**.**render**(**'photos/时间轴图像/中国总体疫情变化.html'**)**#保存数据

1. **疫情数据可视化及前端部署**
   1. **页面布局设计思路**

在前端页面的部署上，利用了jQuery、layui、EasyUI中现有的页面布局样式对前端页面进行美化，用户浏览疫情数据时可以体验到更加友好、简洁直观的页面呈现。jQuery是一个快速、简洁的JavaScript框架，通过ajax发送异步请求得到后端发送的数据并将之展示在页面。layui 提供了丰富的内置模块，可以通过模块化的方式按需加载。我们还用到了jinja2模板作为展示和数据的传输。在网址首页，我们将疫情表格详情、中国疫情地图、中国地区疫情变化分析、世界疫情趋势以及其他信息作为主导航，每个主导航下有各自对应的副导航栏在页面左侧呈现，用户可以根据自己的需求进行个性化选择，从而浏览到对应页面所显示的疫情数据信息。在页面的右上角显示了疫情数据的实时时间以及疫情数据来源的快捷入口，通过点击可以跳转到疫情数据来源页面。其中，我们制作出的疫情数据可视化及前端部署网站链接为：<http://106.52.252.68/>

* 1. **页面效果展示**
     1. **疫情表格详情页面**

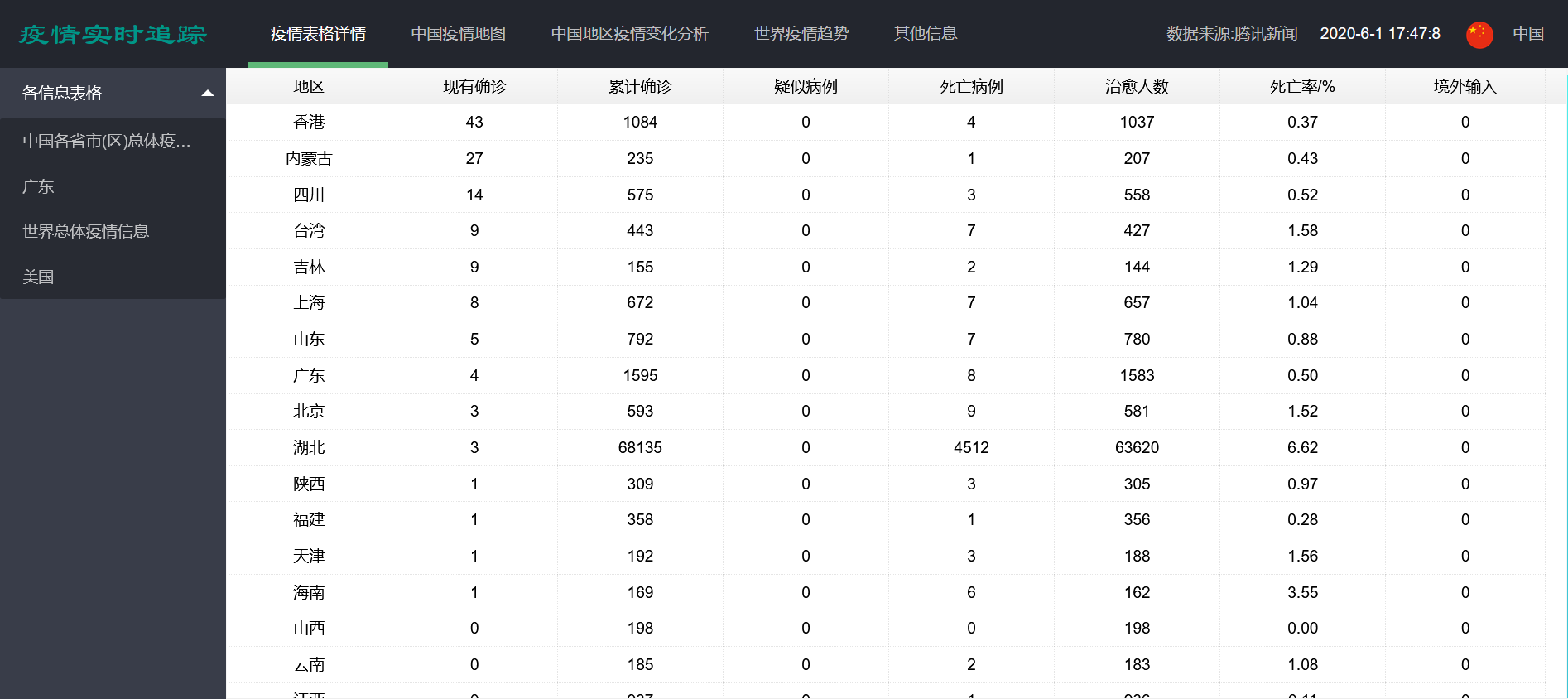


图4-1-1 疫情表格详情页面

在这个页面上，我们将疫情的实时数据以表格的形式呈现给了用户，用户能够直观地得知中国各个省市所对应地现有确诊、累计确诊、疑似病例、死亡病例、治愈病例、死亡率以及境外输入的具体人数。用户还可以通过副导航栏进行选择，从而了解到广东省下属各个城市的实时疫情数据、世界各国总体的疫情信息以及目前累计确诊病例数最多的国家——美国其国内各个城市的疫情信息。

* + 1. **中国疫情地图页面**

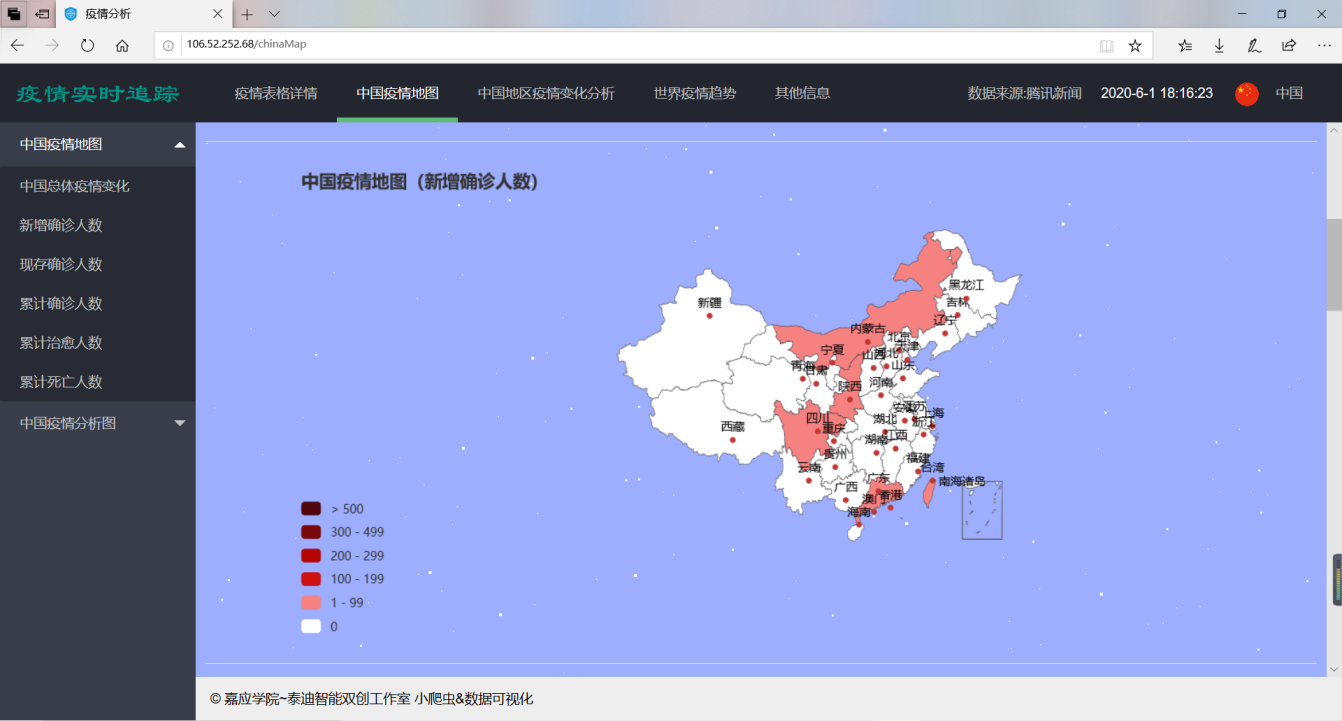


图4-1-2 中国疫情地图页面

此页面展示了中国国内的疫情地图，通过页面左侧的选项卡，用户可以浏览到对应的疫情地图，如，中国总体疫情变化、新增确诊人数、现存确诊人数、累计确诊人数、累计治愈人数以及累计死亡人数。这种疫情地图的可视化使得用户对疫情情况有更直观的体验，通过地图上不同的颜色深度所表现出来的疫情严重程度，中国国内各地区的疫情信息与疫情严重程度差异一目了然。

* + 1. **中国地区疫情变化分析页面**

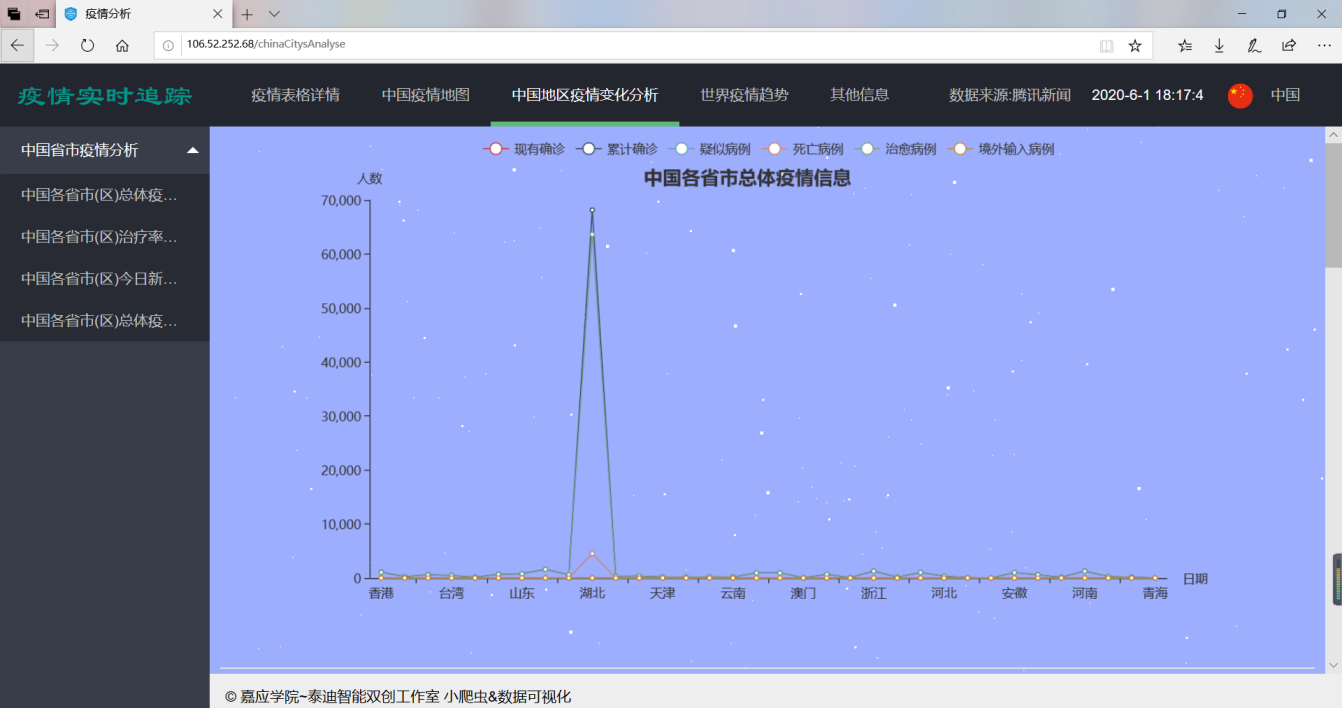


图4-1-3 中国地区疫情变化分析页面

在这个页面，我们以折线图的形式展示了中国各省市总体疫情的变化、治愈率和死亡率。这种简洁的疫情数据表现方式可以让我们迅速地找到一些比较突出的疫情信息，比如，国内累计确诊人数最多的省份——湖北、治愈率最低的省份——吉林、死亡病例数最多的省份——湖北。页面下面的两幅疫情地图也可以让我们了解到中国各省市当天新增病例信息以及总体累计确诊病例信息。

* + 1. **世界疫情趋势页面**

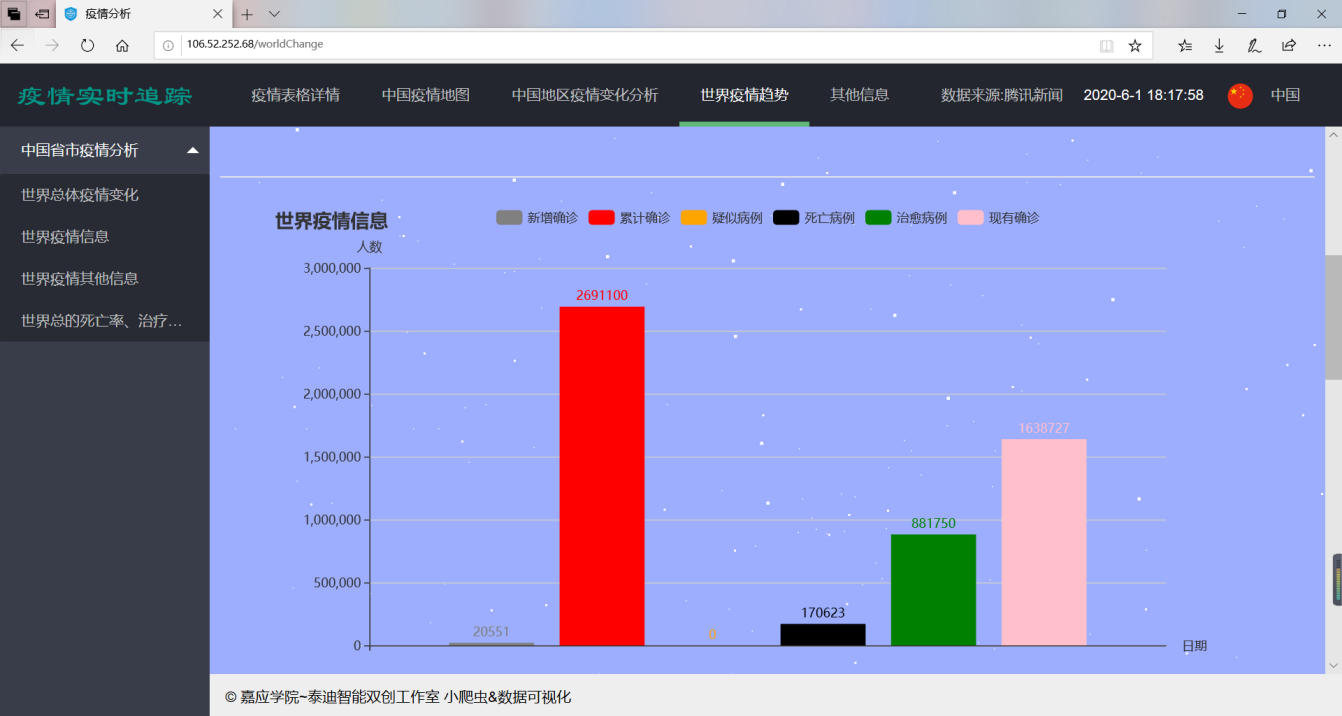


图4-1-4 世界疫情趋势页面

新冠疫情是整个人类的一场灾难，有人道主义责任感的用户会希望了解到全球的新冠疫情状况，以求对世界疫情有一个整体的把握。对此，我们在这个页面上呈现了世界疫情信息的柱状图，从中可以了解到全球实时的新增病例、累计确诊病例、疑似病例、死亡病例、治愈病例以及现有病例。而世界总体的死亡率、治愈率和存活率我们以饼图的形式展示在了页面的下半部分。

* + 1. **其他信息页面**



图4-1-5 其他信息页面

为了满足不同用户的不同需求，在其他信息页面我们展示了一些值得关注的疫情信息。如，各大洲的疫情信息、疫情最严重的十个国家。以洲来划分全球的疫情状况，可以让用户看到更大层面上的疫情分布情况以及各大洲的不同地理环境特点所体现出来的对疫情传播的影响。而疫情严重程度前十国家的折线图和柱状图体现的则是全球疫情核心的迁移情况，如第一波疫情的核心——中国湖北武汉、第二波疫情的核心——美国，在得知这些重要信息的情况下，世界各国可以更好地预防第三波疫情的爆发，从而使新冠疫情在全球范围内逐步得到控制。

1. **总结**

本文利用腾讯新闻平台的公开的数据源，利用Python爬虫技术以及采用可视化工具对这次疫情的数据即使追踪。将这次疫情中体现的世界乃至全中国各个省份的现有确诊、累计确诊、累计治愈、等相关数据进行可视化，以一种更加直观的方式展示出来。与此同时，我们运用jQuery、layui、EasyUI中现有的页面布局样式对前端页面进行美化，将数据以一种更加友好、简洁直观的方式部署到前端页面。

**参考文献**

[1]周冉冉.大数据时代门户网站数据新闻可视化探究 ［D］.山东师范大学，2015. [2]郭晓科.数据新闻学的发展现状与功能［J］.编辑之 友，2013（08）：87-89.

[3]高巍,孙盼盼,李大舟.基于Python爬虫的电影数据可视化分析[J].沈阳化工大学学报,2020,34(01):73-78.

[4]卢星.Python爬虫技术的特性及其应用分析[J].中国新通信,2019,21(15):112-113. [5]周娴.数据新闻在新冠肺炎疫情期间的特点和应用[J].新闻研究导刊,2020,11(07):11+100.

[6]李文满,苏宁,尚美云,李博文,郭利芳,王柳,魏昆,孙献歌,任晓东,任赛,黄庆.新型冠状病毒肺炎的国内研究现状可视化分析与解读[J].国际检验医学杂志,20260,41(09):1025-1029.

[7]叶文.Python语言基于网络学习的数据分析及可视化初探[J].轻工科技,2020,36(05):57-58.

[8]罗仁芝.基于大数据时代下的可视化方法分析[J].信息通信,2020(03):186-188