



江西财经大学  
JIANGXI UNIVERSITY OF FINANCE AND ECONOMICS

课程名称: Python语言与数据分析

## 课 程 报 告

项目名称 新冠肺炎疫情数据分析与可视化

班 级 金融202

学 号 0204820

姓 名 林黄奕

任课教师 肖 泉

开课学期: 2020至2021学年 第二学期

完成时间: 2021年7月3日

---

# 《新冠肺炎疫情数据分析与可视化》数据分析报告

## 目录

1 概述.....	3
(1) 研究背景.....	3
(2) 研究意义.....	3
2 数据描述.....	3
(1) 数据来源.....	3
(2) 数据处理.....	3
(3) 数据特点.....	5
(4) 误差分析.....	5
3 数据分析内容.....	5
4 数据分析图表.....	6
5 数据分析结果.....	12
(1) 全球疫情总量数据.....	12
(2) 全球累计确诊数据.....	12
(3) 全球累计确诊增量数据 .....	12
(4) 各州累计确诊数据.....	12
(5) 各州累计确诊增量数据 .....	13
(6) 中国累计确诊数据.....	13
(7) 中国累计确诊增量数据 .....	13
(8) 中国境外输入数据.....	13
(9) 中国疫情总量数据.....	14
(10) 近 10 天中国累计确诊增量和近十天全世界累计确诊增量 .....	14
(11) 启示.....	14
6 总结.....	15
附录-数据分析代码 .....	15

---

## 1 概述

### (1) 研究背景

2019 年 12 月以来，湖北省武汉市持续开展流感及相关疾病监测，发现多起病毒性肺炎病例，均诊断为病毒性肺炎/肺部感染。新型冠状病毒疫情是国际关注的突发公共卫生事件，已经构成一次全球性“大流行”。

习近平作出重要指示，“各级党委和政府及有关部门要把人民群众生命安全和身体健康放在第一位，制定周密方案，组织各方力量开展防控，采取切实有效措施，坚决遏制疫情蔓延势头。”

### (2) 研究意义

新冠肺炎疫情数据的获取、统计和可视化分析可以让从数据层面更清晰深入地了解新冠疫情在不同国家，不同省份的进展，为遏制疫情提供了依据，有利于采取切实有效措施，对疫情防控工作进行全面的部署，坚决做好疫情防控各项工作。

## 2 数据描述

### (1) 数据来源

来自国家卫健委、各省市区卫健委、各省市区政府、港澳台官方渠道公开数据；

### (2) 数据处理

a.从 <https://voice.baidu.com/act/newpneumonia/newpneumonia> 中爬取国内国外数据并保存至 Excel 中；

b.从 Excel 中读取国外数据并进行整理

---

数据框连接得所有国家的疫情数据；

通过统计缺失值个数来判断是否有缺失值；

对 `gezhou1` 数据框进行数据整理，删除‘热门’等不需要分析的列，得 `gezhou2`

通过 `df.reset_index()` 将索引列变为数据列，得 `gezhou3`；

将各州数据的各列求和；

对 `gezhou3` 中的 `nan` 缺失值进行处理后将‘洲’设为索引，得 `gezhou4`；

国外疫情数据可视化

c. 从 Excel 中读取国内数据并进行整理

对 `zhongguo` 数据框进行数据整理：

对 `zhongguo` 各列求和，得 `zhongguo1`；

通过 `df.reset_index()` 将索引列变为数据列，得 `zhongguo2`；

对 `zhongguo2` 中的 `nan` 缺失值进行处理后通过 `df.set_index()` 重新设置索引，得 `zhongguo3`

国内疫情数据可视化

d. 从连续十天获取的国内国外数据 Excel 表格中获取数据进行整理

读取列表后选取‘累计确诊增量’这一列，并沿横向合并得列表 `jin` 和 `glo` 后对列表的列名修改为相对于的日期。

绘制近十天中国累计确诊增量折线图和近十天全世界累计确诊增量折线图。

---

### (3) 数据特点

数据基本上包含有'累计确诊','死亡','治愈','累计确诊增量','死亡增量','治愈增量','无症状感染者增量','无症状感染者数量','现有确诊','现有确诊增量','境外输入病例'等字段并在此基础上进行增添‘总和’和删减不需要进行分析的数据列。

### (4) 误差分析

从该网站中爬取的数据存在实时更新的误差。每日上午全国数据会优先使用国家卫健委公布的数据，此时各省市数据尚未及时更新，会出现全国数据大于各省份合计数的情况；实时更新全国、各省市数据，因需要核实计算，与官方发布的时间相比，将有一定的时间延迟；国外疫情数据因追踪、核实需要，与各国官方的发布时间相比较有一定的延迟。

## 3 数据分析内容

对全球疫情总量、全球累计确诊、全球累计确诊增量、各州累计确诊病例、各州累计确诊增量、中国疫情总量、中国累计确诊、中国累计确诊增量、中国境外输入病例等内容进行分析，从全球视角着眼简要分析疫情总体分布与不同地域的增长情况后，再将重点放在中国各省份的累计确诊增量和境外输入病例上，分析近 10 天中国累计确诊增量变化，展示疫情发展变化情况与区域差异。

## 4 数据分析图表

图（1）

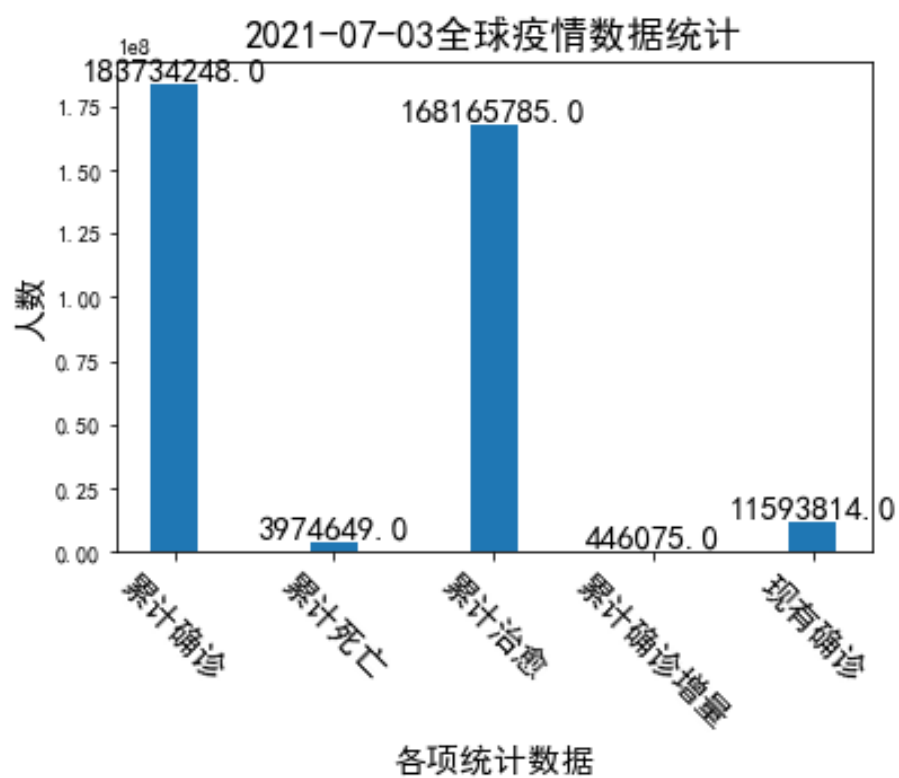


图 (2)

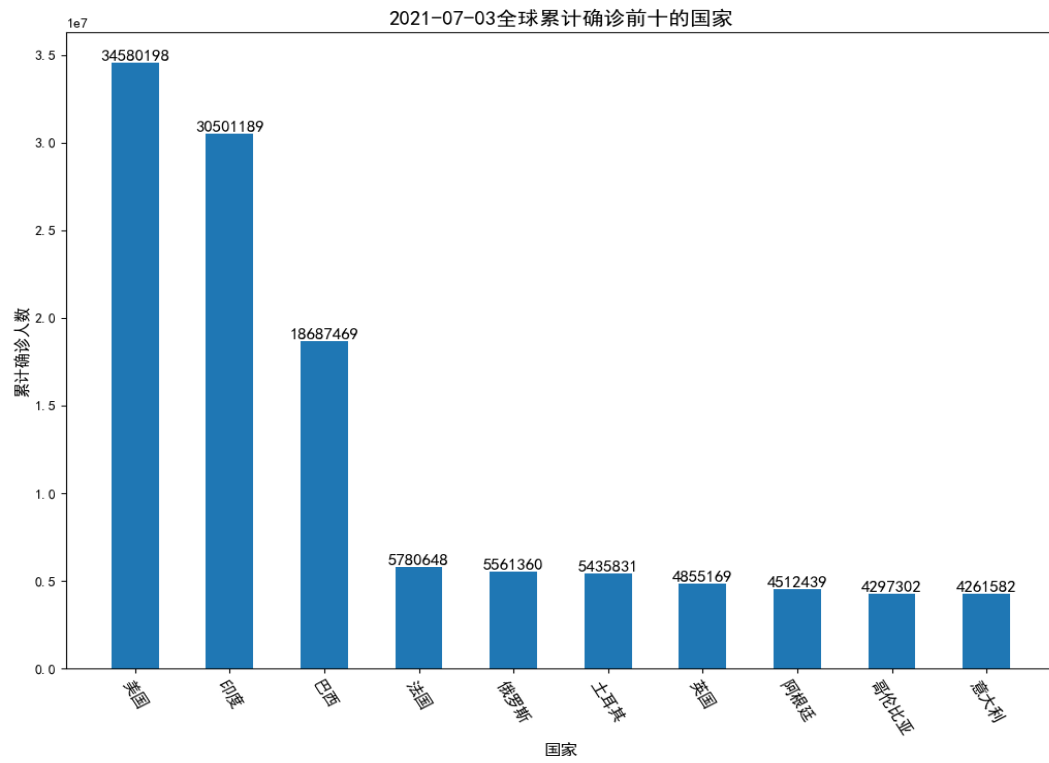
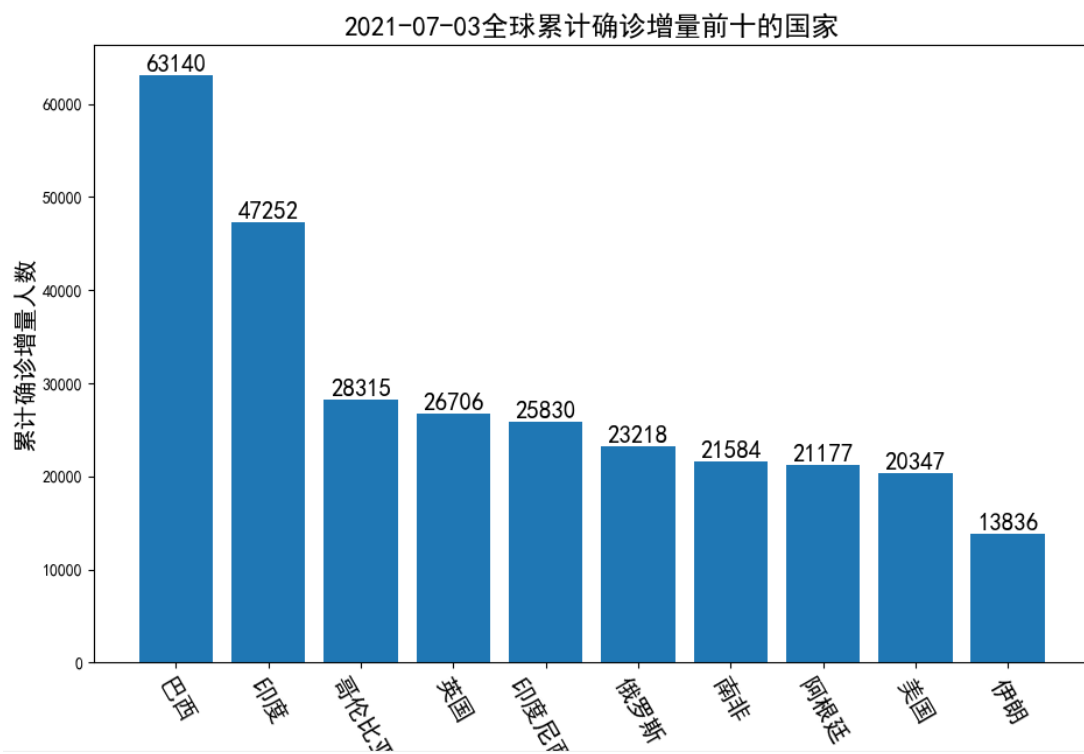
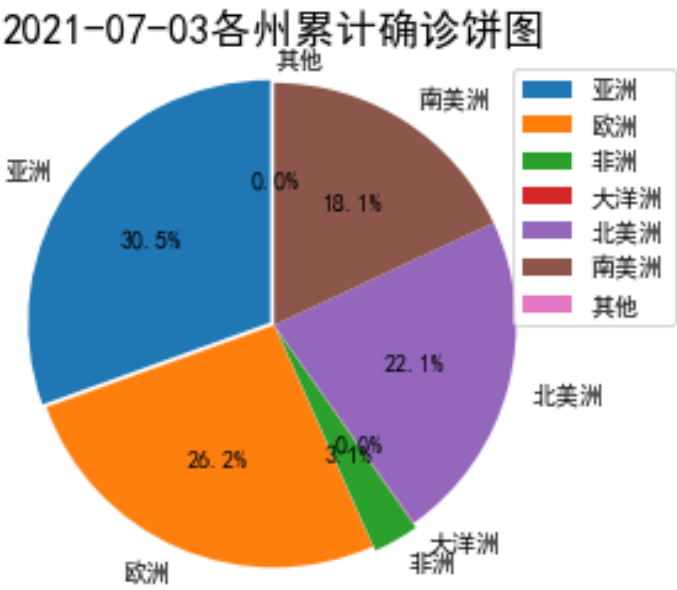


图 (3)



图（4）



图（5）

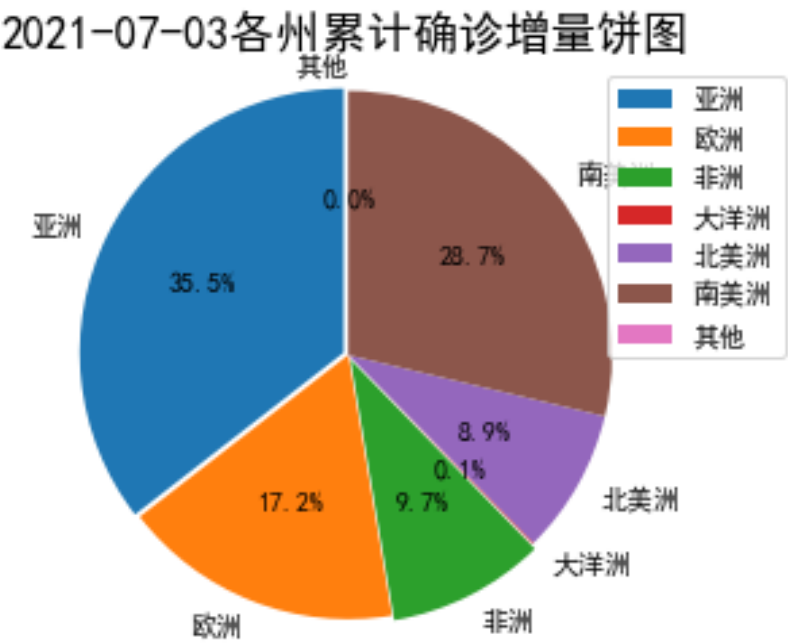




图 (6)

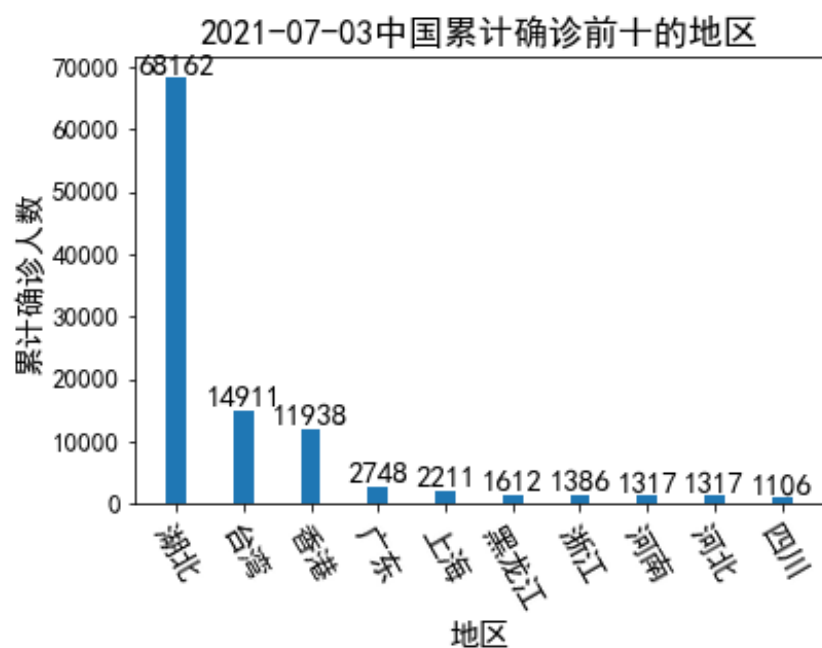


图 (7)

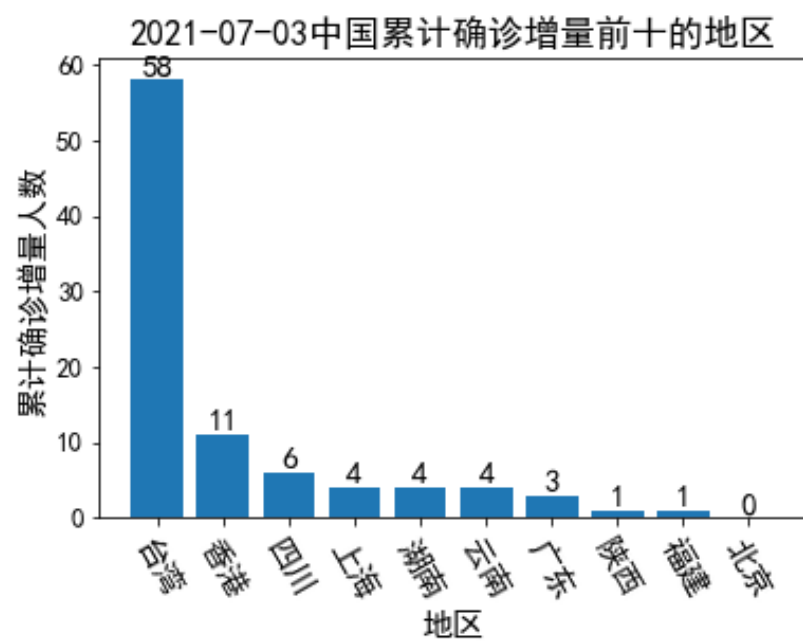


图 (8)

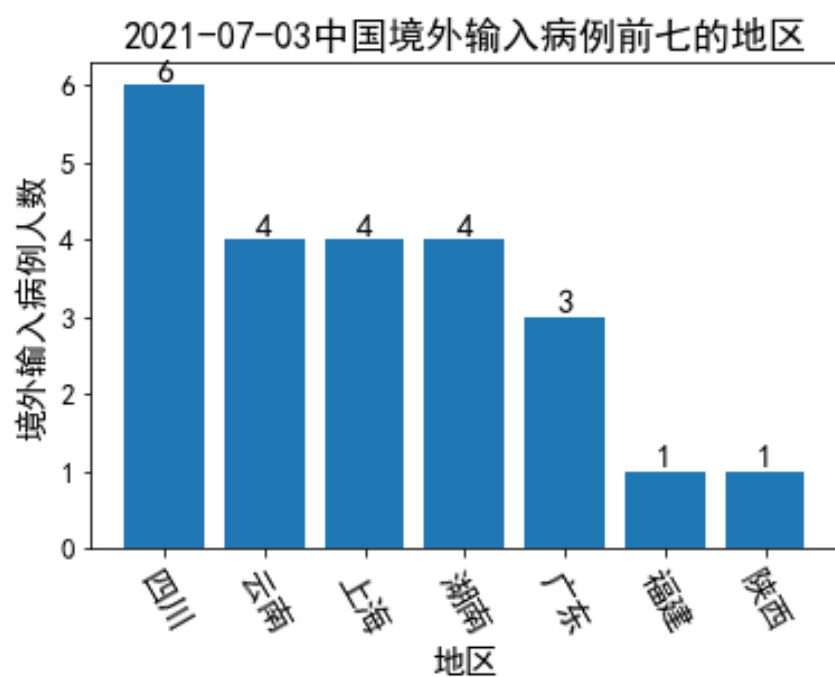


图 (9)

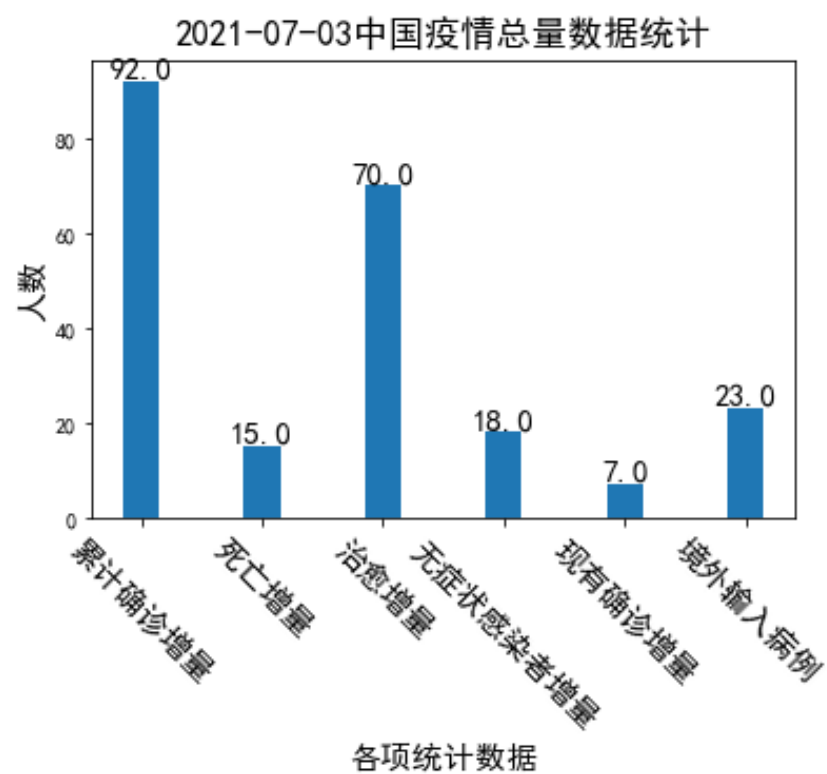


图 (10)

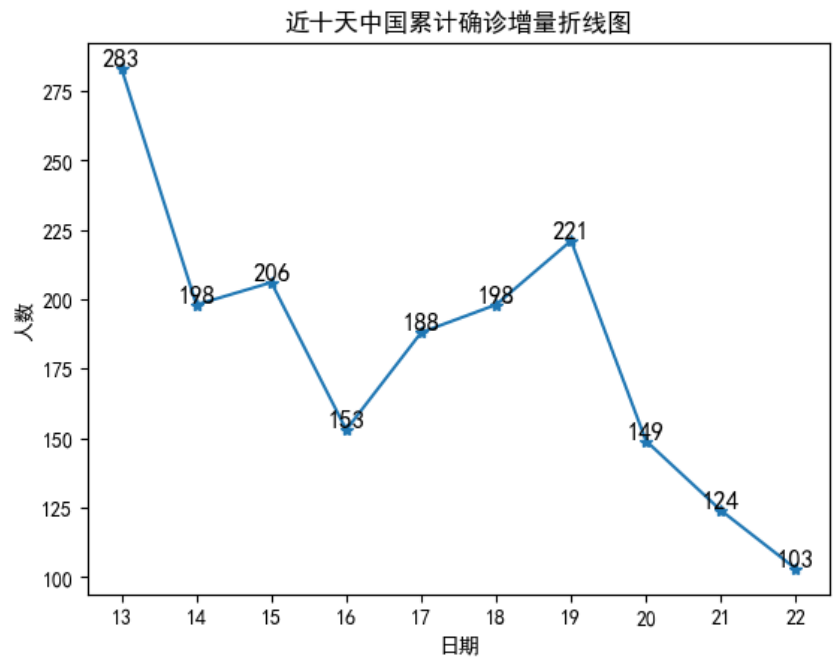
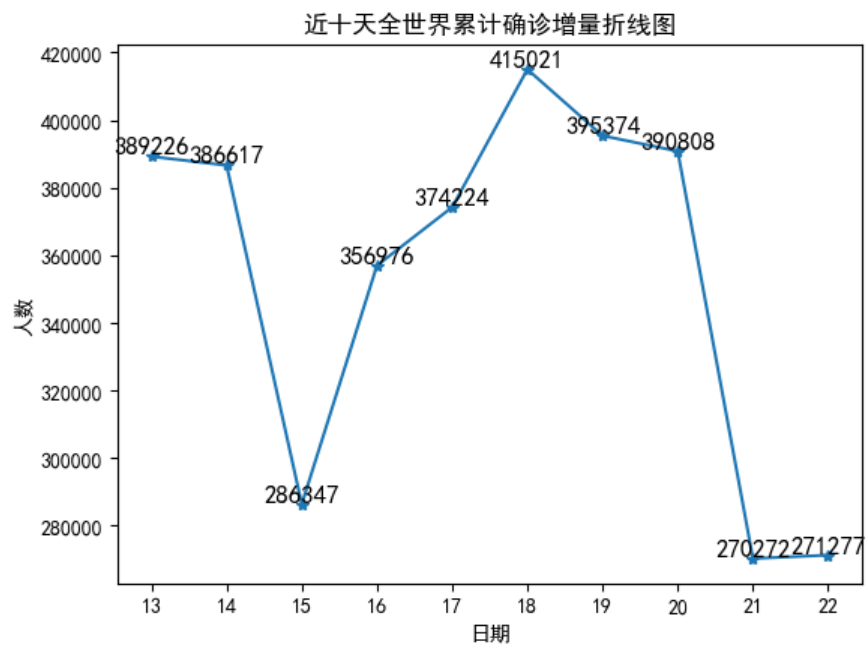


图 (11)



---

## 5 数据分析结果

### (1) 全球疫情总量数据

据百度数据所得的图 (1)，截至 2021 年 7 月 3 日，现有确诊 11593814；累计死亡 3974649；累计确诊 183734248，从中可以得知新冠肺炎疫情死亡人数极高，疫情对全球造成了不可磨灭的伤害，而这正是进行新冠肺炎疫情数据分析与可视化的意义所在。

### (2) 全球累计确诊数据

由图 (2) 可以得知，截至 2021 年 7 月 3 日，新冠疫情在美国造成人员的感染最为严重，其次是印度、巴西、法国等国。从此图可推出，在一定意义上，新冠疫情爆发以来对不同国家国民造成生命安全和身体健康以及整个国家经济等各方面情况的大致的影响程度。

### (3) 全球累计确诊增量数据

由图 (3) 可知，在 2021 年 7 月 3 日，巴西的新冠疫情的累计确诊增量最多，其次是印度、哥伦比亚、英国、印度尼西亚、俄罗斯、南非、阿根廷、美国和伊朗。这说明了当今新冠疫情在巴西最为严重，其次是印度、哥伦比亚等国家。

因此我们要格外注意这几个国家与中国之间的人员流动，包括中国人去往国外和外国人进入中国两种情况，给我们在预防境外输入病例时提供了一定的参考依据。

### (4) 各州累计确诊数据

从图 (4) 可以直观得出截至 2021 年 7 月 3 日，新冠疫情在不

---

同洲的影响程度。亚洲受到新冠疫情的波及最大，其次是欧洲、北美洲、南美洲，非洲受到疫情的波及较小。疫情对于大洋洲的影响最小。

#### **(5) 各州累计确诊增量数据**

从图（5）可以直观得出，在 2021 年 7 月 3 日，亚洲的疫情最为严重，其次是南美洲，欧洲、非洲、北美洲，大洋洲现在受到疫情的影响最小。

#### **(6) 中国累计确诊数据**

从图（6）可知，截至 2021 年 7 月 3 日，新冠疫情爆发以来对我国不同省份造成生命安全和身体健康以及经济等各方面情况的大致的影响程度。新冠疫情对湖北造成的影响最大，其次是台湾、香港，对于广东、上海、黑龙江、浙江、河南、河北、四川等省份也有较大影响。

#### **(7) 中国累计确诊增量数据**

从图（7）可知，在 2021 年 7 月 3 日，疫情在台湾最为严重，其次是香港、四川，在上海、湖南、云南、广东、陕西等省份仍有确诊病例存在，说明对于台湾而言必须加强对疫情的追踪防控。特别地，在香港、四川、上海、湖南、云南、广东、陕西、福建仍旧存在疫情的威胁，不能放松警惕，要做好防疫措施与宣传普及工作。

#### **(8) 中国境外输入数据**

从图（8）可以得知，四川的境外输入病例最多，其次是云南、

---

上海、湖南、广东、福建、陕西。我们在这几个省份要格外注意对境外输入人员的排查。

### **(9) 中国疫情总量数据**

从图（9）可知，中国的累计确诊增量为 92，死亡增量为 15，无症状感染者增量为 18，境外输入病例为 23。

### **(10) 近 10 天中国累计确诊增量和近十天全世界累计确诊增量**

由图（10）可知，在 2021 年 6 月 13 日至 2021 年 6 月 22 日期间，中国累计确诊增量总体上呈下降趋势，这表明中国的疫情在现阶段已经得到了有效的遏制。

由图（11）可知，世界的疫情波动较大，每日的确诊增量数目仍较大，从一定意义上说，全球的经济的发展受到限制，疫情对全球人民的生命安全仍存在威胁。从中可以推出，相较之全世界，中国的疫情防控工作取得较好的成效，居民生活逐渐恢复正常，经济逐渐得到恢复。

### **(11) 启示**

尽管中国已经有效控制住疫情的蔓延，确诊病例增量整体呈下降趋势，但全球的疫情形势不容乐观，我们应该采取相应举措，对境外输入人员进行严密排查，在个别境外输入病例较多的省份更要加大管理力度。但在全世界疫情的严峻形势下，中国不能独善其身，要在做好本国防控工作的同时积极主动地去帮助世界上其他疫情严重的国家，与其他国家守望相助，共克时艰。

---

## 6 总结

在整个数据分析过程中，我虽然遇到了很多难题，却也提升了自主学习、分析问题、动手实践和解决问题的能力。

首先，在参考相关资料后，我对爬虫数据分析有了初步的了解。其次进一步加深了对 Excel 的数据导入与保存以及之后的数据选择合并，排序筛选等整理数据操作的实践运用，并进一步熟悉了进行数据可视化中的 matplotlib 图表处理流程。

我认为在整个研究分析过程中，最重要的是在编写代码之前对选取哪些数据，如何对数据进行处理后并且用哪种图表表示整理出来的数据进行思考，整理好思路后再进行编写，编写时也要注意对相应的代码作相应的注释，这样便于后续对代码的读取与检查。

## 附录-数据分析代码

```
import requests
```

```
import json
```

```
from lxml import etree
```

```
from openpyxl import Workbook
```

```
import pandas as pd
```

```
from datetime import datetime
```

```
import matplotlib
```

```
import matplotlib.pyplot as plt
```

```
#-----
```

```
-
```

---

```
#-----获取数据-----
```

```
-----
```

```
#-----
```

```
-
```

```
url = "https://voice.baidu.com/act/newpneumonia/newpneumonia"
```

```
response = requests.get(url)
```

```
html = etree.HTML(response.text)
```

```
result = html.xpath('//script[@type="application/json"]/text()')
```

```
result = result[0]
```

```
result = json.loads(result)
```

```
#连续十天获取国内数据并保存至 Excel 中
```

```
result1 = result['component'][0]['caseList']
```

```
# 创建工作簿
```

```
wb1 = Workbook()
```

```
# 创建工作表
```

```
ws1 = wb1.active
```

```
ws1.title = "国内疫情"
```

```
ws1.append(['省份','累计确诊','死亡','治愈','累计确诊增量','死亡增量','治愈增量
```

```
','无症状感染者增量','无症状感染者数量','现有确诊','现有确诊增量','境外输入病
```

```
例'])
```



---

```

for each in result1:

    temp_list = [each['area'], each['confirmed'], each['died'], each['crued'],

each['confirmedRelative'], each['diedRelative'], each['curedRelative'],

each['asymptomaticRelative'],each.get('asymptomatic'),each['curConfirm'],

each['curConfirmRelative'], each['overseasInputRelative']]    #each.get() 若数据为

空，值即为 0

    for i in range(len(temp_list)):

        if temp_list[i] == "":

            temp_list[i] = '0'

    ws1.append(temp_list)

wb1.save('./domesticdata.xlsx')

#连续十天获取国外疫情数据并保存至 Excel 中

result2 = result['component'][0]['globalList']

# 创建工作簿

wb2 = Workbook()

# 创建工作表

ws2 = wb2.active

ws2.title = "各州疫情"

ws2.append(['洲','累计确诊','累计死亡','累计治愈','累计确诊增量','现有确诊'])

t = datetime.strftime(datetime.today(),'%Y-%m-%d')    #获取当日日期，并转换成字

符串类型

```

---

```

for each in result2:

    temp_list = [each['area'], each['confirmed'], each['died'], each['crued'],
each['confirmedRelative'], each['curConfirm']]

    for i in range(len(temp_list)):          #confirmed: 累计确诊;
confirmedRelative: 累计确诊增量; curConfirm: 现有确诊

        if temp_list[i] == "":

            temp_list[i] = '0'

    ws2.append(temp_list)

for each in result2:

    sheet_title = each['area']

    # 创建新的工作表

    ws3 = wb2.create_sheet(sheet_title)

    ws3.append(['国家', '累计确诊', '累计死亡', '累计治愈', '累计确诊增量', '现有确
诊'])

    for country in each['subList']:

        temp_list = [country['country'], country['confirmed'], country['died'],
country['crued'], country['confirmedRelative'], country['curConfirm']]

        ws3.append(temp_list)

wb2.save('./globaldata.xlsx')

```

---

```
#-----  
#-----从 Excel 中读取国外数据并进行分析-----  
-----  
#-----  
  
gezhou = pd.read_excel('globaldata.xlsx',index_col=0)           #默认读 Excel 文  
件第一个工作表 ‘各州疫情’  
  
ouzhou = pd.read_excel('globaldata.xlsx',sheet_name='欧洲')     #分别读取各州数  
据存至数据框  
  
yazhou = pd.read_excel('globaldata.xlsx',sheet_name='亚洲')  
  
feizhou = pd.read_excel('globaldata.xlsx',sheet_name='非洲')  
  
dayang = pd.read_excel('globaldata.xlsx',sheet_name='大洋洲')  
  
beimei = pd.read_excel('globaldata.xlsx',sheet_name='北美洲')  
  
nanmei = pd.read_excel('globaldata.xlsx',sheet_name='南美洲')  
  
#-----数据框连接得所有国家的疫情数据-----  
-----  
  
df = pd.concat([yazhou,ouzhou,nanmei,beimei,feizhou,dayang],ignore_index = True)  
  
#通过 ignore_index=True 舍弃原数据框中的索引  
  
#通过统计缺失值个数来判断是否有缺失值  
  
print(df.isnull().sum())  
  
#-----对 gezhou 数据框进行数据整理-----
```

---

-----

#将得到的各州数据去掉'热门',因为'热门',在这里不需要分析

```
gezhou1 = gezhou.drop('热门')
```

```
gezhou2 = gezhou1.reset_index()          #通过 df.reset_index()将索引列变为数据  
列.
```

#将各州数据的各列求和的总量

```
dct = {}
```

```
for i in gezhou1.columns:
```

```
    dct[i] = gezhou1[i].sum()
```

```
gezhou3 = gezhou2.append(dct,ignore_index=True)
```

#对 gezhou3 中的 nan 缺失值进行处理

```
print(gezhou3.isnull())                #判断 gezhou3 中是否是 nan 值,是 nan 值返  
回 True
```

```
gezhou4 = gezhou3.fillna('总和')
```

```
gezhou4 = gezhou4.set_index('洲')      #返回的新数据框将洲列设为索引;gezhou4
```

为数据整理后的各州数据

```
gezhou5 = gezhou4.copy()
```

```
t = datetime.strftime(datetime.today(),'%Y-%m-%d')
```

```
lst = [t]*len(gezhou4.index)
```

```
gezhou5['日期'] = lst
```

---

```
gezhou5.to_excel(t+'global.xlsx')
```

```
#国外疫情数据可视化
```

```
plt.rcParams['font.sans-serif'] = ['SimHei']    # 指定中文黑体字体
```

```
plt.rcParams['axes.unicode_minus'] = False      # 确保-负号显示正常
```

```
#-----获取全球总量数据-----
```

```
----
```

```
plt.figure()
```

```
plt.bar(gezhou4.columns,gezhou4.loc['总和'],width=0.3)
```

```
plt.title(t+'全球疫情数据统计',fontsize=17)
```

```
plt.ylabel("人数",fontsize=15)
```

```
plt.xlabel("各项统计数据",fontsize=15)
```

```
plt.xticks(gezhou4.columns, rotation=-45,fontsize=15)
```

```
for a, b in zip(gezhou4.columns,gezhou4.loc['总和']):
```

```
    plt.text(a, b, b, ha='center', va='bottom',fontsize=15)
```

```
#-----获取全球累计确诊前十的国家并绘制柱形图-----
```

```
-----
```

```
plt.figure()
```

```
df1 = df.sort_values(by = '累计确诊',ascending=False)
```

```
df11 = df1.head(10)    #通过 df.head(n)获取前十的数据
```

---

```
plt.bar(df11.国家,df11.累计确诊,width=0.5)

plt.title(t+'全球累计确诊前十的国家',fontsize=15)

plt.ylabel("累计确诊人数",fontsize=12)

plt.xlabel("国家",fontsize=12)

plt.xticks(list(df11.国家), rotation=-60,fontsize=12)

for a, b in zip(df11.国家,df11.累计确诊):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=12)


#-----获取全球累计确诊增量前十的国家并绘制柱形图-----
-----

plt.figure()

df2 = df.sort_values(by = '累计确诊增量',ascending=False)

df21 = df2.head(10)    #通过 df.head(n)获取前十的数据

plt.bar(df21.国家,df21.累计确诊增量)

plt.title(t+'全球累计确诊增量前十的国家',fontsize=17)

plt.ylabel("累计确诊增量人数",fontsize=15)

plt.xlabel("国家",fontsize=15)

plt.xticks(list(df21.国家), rotation=-60,fontsize=15)

for a, b in zip(df21.国家,df21.累计确诊增量):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=15)
```

---

```
#-----绘制各州累计确诊饼图-----

plt.figure()

labels = gezhou2.洲

rate = gezhou2.累计确诊

explode = (0.02,0,0.02,0,0,0,0)

patches,ltext,ptext = plt.pie(rate,explode=explode,labels=labels,

                                autopct='%1f%%', shadow=False, startangle=90)

plt.axis('equal')

plt.title(t+'各州累计确诊饼图',fontsize=17)

matplotlib.rcParams.update({'font.size': 10})

plt.legend()


#-----绘制各州累计确诊增量饼图-----

plt.figure()

labels = gezhou2.洲

rate = gezhou2.累计确诊增量

explode = (0.02,0,0.02,0,0,0,0)

patches,ltext,ptext = plt.pie(rate,explode=explode,labels=labels,

                                autopct='%1f%%', shadow=False, startangle=90)
```

---

```
plt.axis('equal')

plt.title(t+'各州累计确诊增量饼图',fontsize=17)

matplotlib.rcParams.update({'font.size': 10})

plt.legend()


#-----

#-----从 Excel 中读取国内数据并进行分析-----

-----

#-----

#-----对 zhongguo 数据框进行数据整理-----

-----

zhongguo = pd.read_excel('domesticdata.xlsx',index_col=0)

#对 zhongguo 各列求和的全国数据

zhongguo1 = zhongguo.reset_index()          #通过 df.reset_index()将索引列变为
数据列,将索引设置为 0,1,2,3……

dct = {}

for i in zhongguo.columns:

    dct[i] = zhongguo[i].sum()

zhongguo2 = zhongguo1.append(dct,ignore_index=True)

#对 zhongguo2 中的 nan 缺失值进行处理
```



---

```

print(zhongguo2.isnull())          #判断 zhongguo2 中是否是 nan 值,是 nan
值返回 True

zhongguo3 = zhongguo2.fillna('总和')

zhongguo3 = zhongguo3.set_index('省份')    #通过 df.set_index() 重新设置索引

zhongguo4 = zhongguo3.copy()

lst2 = [t]*len(zhongguo4.index)

zhongguo4['日期'] = lst2

zhongguo4.to_excel(t+'domestic.xlsx')


#国内疫情数据可视化

#-----获取中国累计确诊前十的地区并绘制柱形图-----
-----

plt.figure()

z1 = zhongguo.sort_values(by = '累计确诊',ascending=False)

z11 = z1.head(10)    #通过 df.head(n)获取前十的数据

plt.bar(z11.index,z11.累计确诊,width=0.3)

plt.title(t+'中国累计确诊前十的地区',fontsize=17)

plt.ylabel("累计确诊人数",fontsize=15)

plt.xlabel("地区",fontsize=15)

plt.xticks(list(z11.index), rotation=-60,fontsize=15)

for a, b in zip(z11.index,z11.累计确诊):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=15)

```

---

#-----获取中国累计确诊增量前十的地区并绘制柱形图-----

-----

```
plt.figure()

z2 = zhongguo.sort_values(by = '累计确诊增量',ascending=False)

z21 = z2.head(10)    #通过 df.head(n)获取前十的数据

plt.bar(z21.index,z21.累计确诊增量)

plt.title(t+'中国累计确诊增量前十的地区',fontsize=17)

plt.ylabel("累计确诊增量人数",fontsize=15)

plt.xlabel("地区",fontsize=15)

plt.xticks(list(z21.index), rotation=-60,fontsize=15)

for a, b in zip(z21.index,z21.累计确诊增量):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=15)
```

#-----获取中国境外输入病例前七的地区并绘制柱形图-----

-----

```
plt.figure()

z3 = zhongguo.sort_values(by = '境外输入病例',ascending=False)

z31 = z3.head(7)    #通过 df.head(n)获取前八的数据

plt.bar(z31.index,z31.累计确诊增量)
```

---

```

plt.title(t+'中国境外输入病例前七的地区',fontsize=17)

plt.ylabel("境外输入病例人数",fontsize=15)

plt.xlabel("地区",fontsize=15)

plt.xticks(list(z31.index), rotation=-60,fontsize=15)

for a, b in zip(z31.index,z31.累计确诊增量):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=15)


#-----获取中国总量数据-----
----

plt.figure()

zhongguo5 = zhongguo3.drop(['治愈','累计确诊','死亡','现有确诊','无症状感染者数
量'],axis=1)

plt.bar(zhongguo5.columns,zhongguo5.loc['总和'],width=0.3)

plt.title(t+'中国疫情总量数据统计',fontsize=17)           #排除累计确诊、治
愈，死亡等列后

plt.ylabel("人数",fontsize=15)

plt.xlabel("各项统计数据",fontsize=15)

plt.xticks(zhongguo5.columns, rotation=-45,fontsize=15)

for a, b in zip(zhongguo5.columns,zhongguo5.loc['总和']):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=15)


#-----近十天中国累计确诊增量数据-----

```

---

#读取列表后合并列表

#读取列表

```
j13= pd.read_excel('20210613domestic.xlsx',index_col=0).累计确诊增量
j14= pd.read_excel('2021-06-14domestic.xlsx',index_col=0).累计确诊增量
j15= pd.read_excel('2021-06-15domestic.xlsx',index_col=0).累计确诊增量
j16= pd.read_excel('2021-06-16domestic.xlsx',index_col=0).累计确诊增量
j17= pd.read_excel('2021-06-17domestic.xlsx',index_col=0).累计确诊增量
j18= pd.read_excel('2021-06-18domestic.xlsx',index_col=0).累计确诊增量
j19= pd.read_excel('2021-06-19domestic.xlsx',index_col=0).累计确诊增量
j20= pd.read_excel('2021-06-20domestic.xlsx',index_col=0).累计确诊增量
j21= pd.read_excel('2021-06-21domestic.xlsx',index_col=0).累计确诊增量
j22= pd.read_excel('2021-06-22domestic.xlsx',index_col=0).累计确诊增量
```

#合并列表

```
j1n = pd.concat([j13,j14,j15,j16,j17,j18,j19,j20,j21,j22],axis = 1)

j1n.columns = ['13','14','15','16','17','18','19','20','21','22']
```

#绘制近十天中国累计确诊增量折线图

```
plt.figure()

plt.plot(j1n.columns,j1n.loc['总和'],marker = '*')

plt.title('近十天中国累计确诊增量折线图')

plt.xlabel('日期')

plt.ylabel('人数')
```

---

```
for a, b in zip(jin.columns,jin.loc['总和']):
```

```
    plt.text(a, b, b, ha='center', va='bottom',fontsize=12)
```

```
#-----近十天全世界累计确诊增量数据-----
```

```
---
```

```
#读取列表后合并列表
```

```
#读取列表
```

```
g13= pd.read_excel('2021-06-13global.xlsx',index_col=0).累计确诊增量
```

```
g14= pd.read_excel('2021-06-14global.xlsx',index_col=0).累计确诊增量
```

```
g15= pd.read_excel('2021-06-15global.xlsx',index_col=0).累计确诊增量
```

```
g16= pd.read_excel('2021-06-16global.xlsx',index_col=0).累计确诊增量
```

```
g17= pd.read_excel('2021-06-17global.xlsx',index_col=0).累计确诊增量
```

```
g18= pd.read_excel('2021-06-18global.xlsx',index_col=0).累计确诊增量
```

```
g19= pd.read_excel('2021-06-19global.xlsx',index_col=0).累计确诊增量
```

```
g20= pd.read_excel('2021-06-20global.xlsx',index_col=0).累计确诊增量
```

```
g21= pd.read_excel('2021-06-21global.xlsx',index_col=0).累计确诊增量
```

```
g22= pd.read_excel('2021-06-22global.xlsx',index_col=0).累计确诊增量
```

```
#合并列表
```

```
glo = pd.concat([g13,g14,g15,g16,g17,g18,g19,g20,g21,g22],axis = 1)
```

```
glo.columns = ['13','14','15','16','17','18','19','20','21','22']
```

```
#绘制近十天全世界累计确诊增量折线图
```

---

```
plt.figure()

plt.plot(glo.columns,glo.loc['总和'],marker = '*')

plt.title('近十天全世界累计确诊增量折线图')

plt.xlabel('日期')

plt.ylabel('人数')

for a, b in zip(glo.columns,glo.loc['总和']):

    plt.text(a, b, b, ha='center', va='bottom',fontsize=12)
```

