

# 南昌大学

## 软件学院大作业任务书

题 目：\_\_\_\_\_ 基于新冠肺炎疫情的分析与预测 \_\_\_\_\_

专 业：\_\_\_\_\_ 软件工程 \_\_\_\_\_

班 级：\_\_\_\_\_ 软件工程[1801-14]班 \_\_\_\_\_

学 号：\_\_\_\_\_ 8002118261 8002118240 \_\_\_\_\_

学生姓名：\_\_\_\_\_ 曹蓉 杨孟衡 \_\_\_\_\_

起讫日期：\_\_\_\_\_ 2021.5.10— 2021.6.20 \_\_\_\_\_

任课教师：\_\_\_\_\_ 刘伯成 \_\_\_\_\_ 职称：\_\_\_\_\_ 讲师 \_\_\_\_\_

部分管主任：\_\_\_\_\_ 徐健锋 \_\_\_\_\_

完成时间：\_\_\_\_\_ 2021.6.20 \_\_\_\_\_

## 说 明

1. 本任务书由任课教师填写后，下达到学生。
2. 任务完成后，任课教师需填写小结表。
3. 任务书、学生成绩与学生完成后的大作业（纸质和电子两种）一并报送各教学研究部审核后转教务办。
4. 另附一份全班大作业总结

## 大作业的要求和内容：（包括题目选择范围、技术要求、递交时间、考核方法等）

综合运用所学知识，采用团队合作方式（不超过 4 名）完成一个系统或模型的开发项目，可以是信息管理系统或智能监控模型。

具体来说，信息管理系统可以是员工管理系统、工资管理系统、课程管理系统等等与信息管理系统相关的系统，相关开发可以参考教材第 9 章的购物系统。智能监控模型包括两个可选方向：一是基于视频的水域游泳者检测，即通过对视频进行处理检测水域中是否出现游泳者；二是基于视频的救生衣检测，即通过对视频进行处理检测水域中出现的人是否穿有救生衣。智能监控模型的两个参考案例见附件，可以在它们的基础上做新的开发。信息管理系统相对于智能监控模型在开发上更容易，因此智能监控模型将在评分过程中的选题和代码质量方面具有更高的难度系数分。

大作业的详细内容和功能不做具体限定，但不能过于简单。关于信息管理系统，不得抄袭或从网上下载充当。关于智能监控模型的两个方向，网上没有现成的公开代码，允许在开发过程中复用他人的代码。此外，信息管理系统和智能监控模型必须满足以下要求。

### 技术要求：

- 使用顺序、选择、循环、跳转语句编写程序。
- 使用列表等组合数据类型。
- 定义类、创建和使用对象。
- 使用 Python 解决实际问题。

最终提交的作业，除提交完整的代码外，还必须以大作业报告的形式阐述整个实现过程，具体包括：

- **需求分析**（项目介绍、功能需求，数据获取等）；
- **项目分析与设计**（阐述项目中需解决的关键技术问题，同时要以功能模块示意图等辅助项目设计的描述）；
- **项目设计与实现**（其中内容**不能只是粘贴全部代码**，需要描述清楚算法流程。如果必须给出实现代码才能更好地说明问题时，也必须先有相关的文字叙述，然后才是代码，代码只是作为例证。）；
- **个人小结**（该部分为个人开发小结，其中必须谈到开发过程中遇到的困难以及如果克服困难、个人收获、得到的启示或教训等等，切忌空洞无实际

内容或千篇一律的敷衍文字。))；

- **参考文献**（该部分给出整个项目从选题、需求分析、设计到实现过程中所参考的书籍、网上资料等。))。

大作业的评分点涵盖大作业从选题、需求分析、项目实现到文档撰写全过程。具体评分点及各评分点的比重如下：

- **选题** 20%  
评分依据：选题的难度、创新度、工作量等
- **需求分析、数据采集** 10%  
评分依据：分析是否充分、表述是否明确、功能的实用价值等
- **文档撰写质量** 30%  
评分依据：结构完整性、内容充实度、格式符合度、图表规范程度等
- **代码质量** 40%  
评分依据：代码复杂度、功能完整性、是否运用了要求的知识点、设计或算法是否有创新等

教师小结：

成绩：\_\_\_\_\_

教师签名：\_\_\_\_\_

教研部负责人：\_\_\_\_\_

学生姓名： 曹蓉、杨孟衡

# 南昌大学

NANCHANG UNIVERSITY

## 课程设计报告



题    目: \_\_\_\_\_ 基于新冠肺炎疫情的分析与预测 \_\_\_\_\_

学    院: \_\_\_\_\_ 软件学院 \_\_\_\_\_

班    级: \_\_\_\_\_ 软件工程[1801-14]班 \_\_\_\_\_

学    号: \_\_\_\_\_ 8002118261      8002118240 \_\_\_\_\_

姓    名: \_\_\_\_\_ 曹蓉                  杨孟衡 \_\_\_\_\_

起讫日期: \_\_\_\_\_ 2021. 5. 10— 2021.6. 20 \_\_\_\_\_

任课教师: \_\_\_\_\_ 刘伯成                  职称: \_\_\_\_\_ 讲师 \_\_\_\_\_

部分管主任: \_\_\_\_\_ 徐健锋 \_\_\_\_\_

完成时间: \_\_\_\_\_ 2021.6. 20 \_\_\_\_\_

## 目录

一、 项目需求分析.....	19
1.1 项目介绍.....	19
1.2 功能需求.....	19
二、 项目分析与设计.....	20
2.1 本项目需解决的关键技术问题.....	20
2.2 项目流程.....	20
2.3 功能模块.....	21
三、 项目设计与实现.....	22
3.1 项目设计.....	22
3.1.1 数据集来源.....	22
3.1.2 数据说明.....	22
3.2 项目实施.....	23
3.2.1 项目实施全球新冠疫情的总体变化趋势.....	23
3.2.2 累计确诊数排名前 20 的国家名称及其数量.....	25
3.2.3 日新增确诊数累计排名前 10 的国家.....	26
3.2.4 累计确诊人数占国家总人口比例最高的 10 个国家.....	27
3.2.5 死亡率最低的 10 个国家.....	28
3.2.6 展示各个国家的累计确诊人数的比例.....	29
3.2.7 展示全球各个国家累计确诊人数的箱型图.....	31
3.2.8 康复率最高的 10 个国家.....	32
3.2.9 分析全世界应对新冠疫情最好的 10 个国家.....	33
3.2.10 疫情预测.....	34
四、 设计日志.....	37
4.1 界面无法正常显示.....	37
4.2 无法导入相应包.....	38
五、 个人小结.....	38
六、 参考文献.....	40

## 一、 项目需求分析

### 1.1 项目介绍

新型冠状病毒肺炎（Corona Virus Disease 2019, COVID-19），简称“新冠肺炎”，世界卫生组织命名为“2019 冠状病毒病”，是指 2019 新型冠状病毒感染导致的肺炎。2019 年 12 月以来，湖北省武汉市部分医院陆续发现了多例有华南海鲜市场暴露史的不明原因肺炎病例，证实为 2019 新型冠状病毒感染引起的急性呼吸道传染病。随后病情迅速蔓延，对人们生活的方方面面产生了重要影响，并引发国内外舆论的广泛关注。本文通过分析全球疫情数据，并针对全球累计确诊数，利用前 10 天采集到的数据做后 5 天的预测，并与实际数据进行对比，分析比较与实际数据的差距和原因。

### 1.2 功能需求

本项目对新冠肺炎疫情数据分析需求如下：

- 1) 15 天中，全球新冠疫情的总体变化趋势；
- 2) 累计确诊数排名前 20 的国家名称及其数量；
- 3) 15 天中，每日新增确诊数累计排名前 10 个国家的每日新增确诊数据的曲线图；
- 4) 累计确诊人数占国家总人口比例最高的 10 个国家；
- 5) 死亡率（累计死亡人数/累计确诊人数）最低的 10 个国家；
- 6) 饼图展示各个国家的累计确诊人数的比例；
- 7) 展示全球各个国家累计确诊人数的箱型图；
- 8) 根据分析结果，列出全世界应对新冠疫情最好的 10 个国家；
- 9) 针对全球累计确诊数，利用前 10 天采集到的数据做后 5 天的预测，并与实际数据进行对比，分析比较与实际数据的差距和原因。

## 二、 项目分析与设计

### 2.1 本项目需解决的关键技术问题

- 1) 分析各国新冠肺炎疫情情况，确定影响疫情的关键因素；
- 2) 设置影响新冠肺炎疫情的关键因素的权值；
- 3) 确定预测方法进行疫情预测。

### 2.2 项目流程

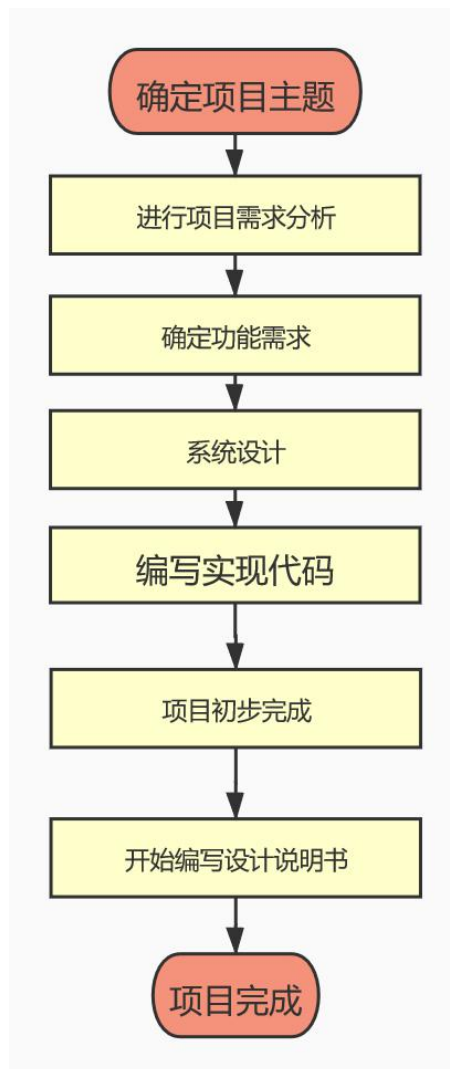


图 1 项目流程图



## 2.3 功能模块

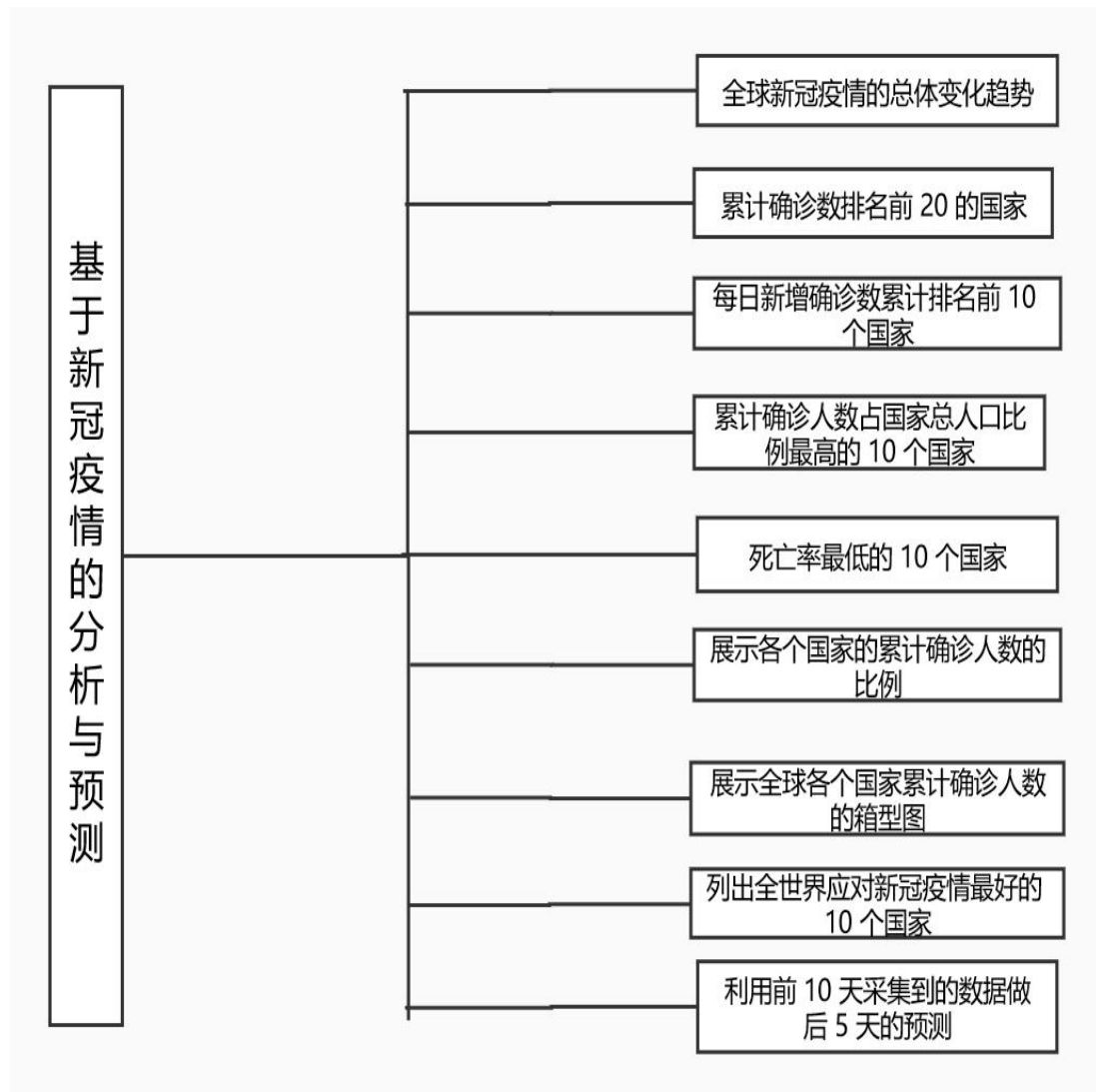


图 2 功能模块图

### 三、 项目设计与实现

#### 3.1 项目设计

##### 3.1.1 数据集来源

使用的数据来自 <https://ncov2019.live/> 网站上世界各国新冠肺炎疫情数据集。

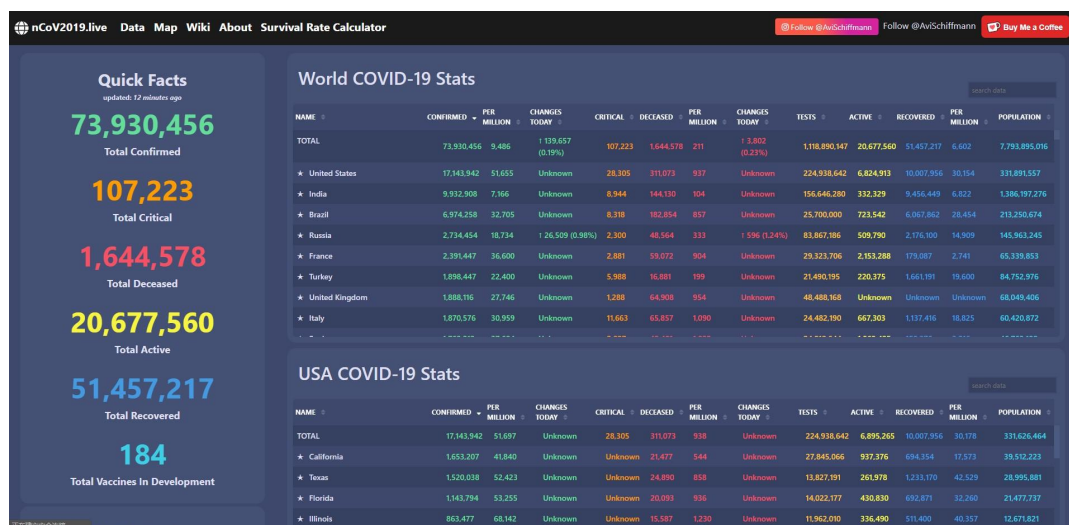


图 3 数据集来源图

##### 3.1.2 数据说明

我们选取了 2020 年 12 月 1 日-12 月 15 日各国每日的疫情数据，具体格式如下：

- 1) 国家名称: Name;
- 2) 累计确诊人数: Confirmed;
- 3) 每百万确诊数: Confirmed Per Million;
- 4) 新增确诊数: Confirmed Changes Today;
- 5) 新增确诊百分比: Confirmed Percentage Day Change;
- 6) 死亡病例: Deceased;
- 7) 每百万死亡数: Deceased Per Million;
- 8) 新增死亡人数: Deceased Changes Today;
- 9) 新增死亡百分比: Death Percentage Day Change;
- 10) 治愈病例: Recovered;

11) 每百万治愈数: Recovered Per Million;

12) 人口数: Population;

	Name	Confirmed	Confirmed Per Million	Confirmed Changes Today	Confirmed Percentage
1	TOTAL	73,252,453	9,399	70,962	0.1%
2	Afghanistan	49,703	1,264	219	0.44%
3	Albania	49,191	17,102	0	0%
4	Algeria	92,597	2,095	0	0%
5	Andorra	7,382	NA	0	0%
6	Angola	16,277	488	0	0%
7	Antigua and Barbuda	148	NA	0	0%
8	Argentina	1,503,222	33,122	0	0%
9	Armenia	149,120	50,280	438	0.29%
10	Austria	325,051	35,998	0	0%
11	Azerbaijan	178,986	17,581	0	0%
12	The Bahamas	7,674	NA	0	0%
13	Bahrain	89,268	51,666	0	0%
14	Bangladesh	492,332	2,976	0	0%
15	Barbados	296	NA	0	0%
16	Belarus	162,148	17,162	0	0%
17	Belize	9,377	NA	0	0%
18	Benin	3,090	252	0	0%
19	Bhutan	438	NA	0	0%
20	Bolivia	147,345	12,545	195	0.13%
21	Bosnia and Herzegovina	101,461	31,014	0	0%

图 4 csv 文件数据格式

## 3.2 项目实施

### 3.2.1 项目实施全球新冠疫情的总体变化趋势

通过分析 15 日全球新冠疫情数据, 得到如下数据分析图:

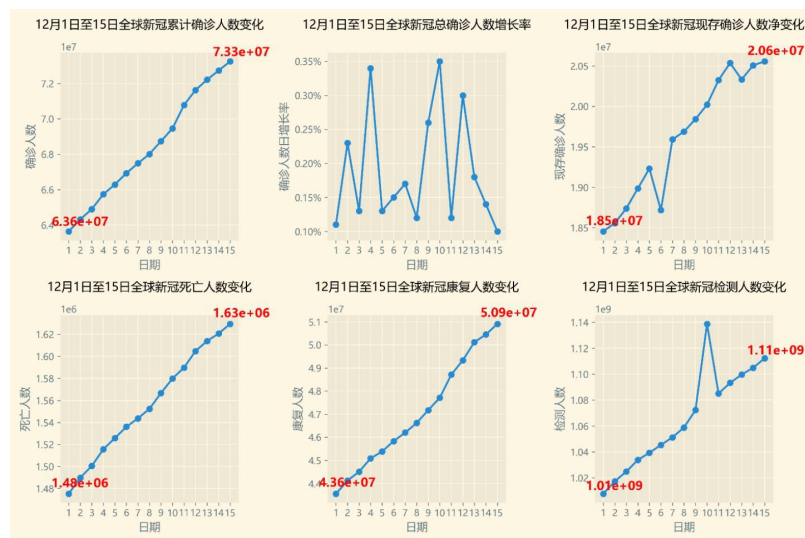


图 5 全球疫情总体变化趋势

可以看出在这 15 日内，全球新冠确诊人数仍然在持续上涨，累计确诊人数从 1 日的 6.36 千万人增长至 7.33 千万人，净确诊人数（总增长人数减去死亡与治愈人数）从 1.85 千万人增长至 2.06 千万人，总确诊人数日增长率维持在 0.10%到 0.35%之间。另外，全球死亡人数、康复人数与检测人数则呈现类似线性的增长趋势上升。

关键代码实现如下：

```
for i in range(1, 16):
    if i >= 10:
        str_num = str(i)
    else:
        str_num = '0' + str(i)
    df[str(i)] = \
        pd.read_csv('csvFile/Covid19Data2020-12-' + str_num + '.csv',
                    encoding='utf-8', thousands=',', nrows=1).loc[0] # 去除千分位的逗号

# 1.展示全球新冠疫情总确认数量变化
Confirmed_list = df.loc['Confirmed'].to_list()
ax[0, 0].set_title('12 月 1 日至 15 日全球新冠累计确诊人数变化', y=1.1, size=13)

# 2.展示全球新冠疫情增长速度变化
Confirmed_Percentage_list = df.loc['Confirmed Percentage Day Change'].to_list()
ax[0, 1].set_title('12 月 1 日至 15 日全球新冠总确诊人数增长率', y=1.1, size=13)

# 3.展示现存确诊人数
Active_list = df.loc['Active'].to_list()
ax[0, 2].set_title('12 月 1 日至 15 日全球新冠现存确诊人数净变化', y=1.1, size=13)

# 4.展示全球新冠疫情死亡人数变化
Deceased_list = df.loc['Deceased'].to_list()
ax[1, 0].set_title('12 月 1 日至 15 日全球新冠死亡人数变化', y=1.1, size=13)

# 5.展示全球新冠疫情康复人数变化
Recovered_list = df.loc['Recovered'].to_list()
ax[1, 1].set_title('12 月 1 日至 15 日全球新冠康复人数变化', y=1.1, size=13)

# 6.展示全球新冠疫情检测人数变化
Tests_list = df.loc['Tests'].to_list()
ax[1, 2].set_title('12 月 1 日至 15 日全球新冠检测人数变化', y=1.1, size=13)
```

图6 关键代码

### 3.2.2 累计确诊数排名前 20 的国家名称及其数量

利用12月15日的数据，分析累计确诊数排名前20的国家名称及其数量，得到如下柱状图：

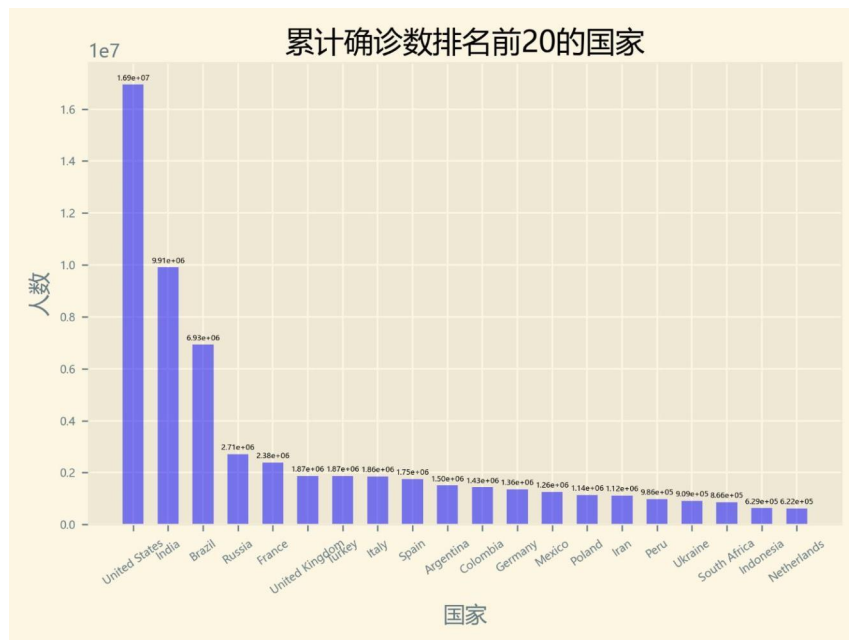


图7 累计确诊数排名前20的国家

其中，美国、印度与巴西的累计确诊人数数据较为突出，分别为1.69千万人、9.91百万人和6.93百万人。而其它所有国家的累计确诊数据均不超过 3 百万人。

关键代码实现如下：

```
plt.bar(list(range(0, 100, 5)), df_res['Confirmed'].to_list(), width=3, alpha=0.5, color='b')
plt.xticks(list(range(0, 100, 5)), labels=df_res['Name'].to_list(), rotation=35)
plt.tick_params(labels=6)
for a, b in zip(list(range(0, 100, 5)), df_res['Confirmed'].to_list()): # 在直方图上显示数字
    plt.text(a, b + 1e5, '%.2e' % b, ha='center', va='bottom', fontsize=4, color='black')
plt.title('累计确诊数排名前 20 的国家')
plt.xlabel("国家")
plt.ylabel("人数")
```

图8 关键代码

### 3.2.3 日新增确诊数累计排名前 10 的国家

分析 15 天中，每日新增确诊数累计排名前 10 的国家，并作从 2 日开始的每日新增确诊数据的曲线图如下所示：

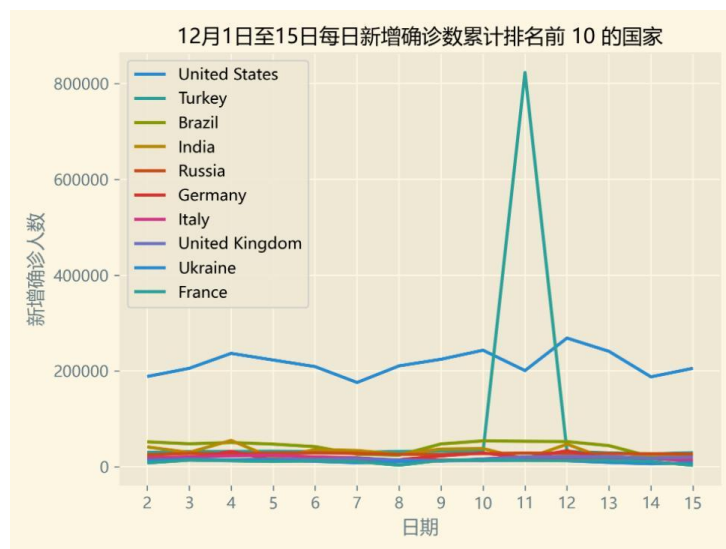


图9 日新增确诊数累计排名前10的国家

由上至下为该排名的先后顺序。美国为每日新增确诊人数最多的国家，其每日新增人数远远超过其它任何国家。然而，在12月11日，土耳其新增确诊人数突然增加，这是由于土耳其疫情反弹，检测人数大量增加导致的。

关键代码实现如下：

```
# 只读取每日确诊数据
for i in range(1, 16):
    if i >= 10:
        str_num = str(i)
    else:
        str_num = '0' + str(i)
    df_temp = pd.read_csv('csvFile/Covid19Data2020-12-' + str_num + '.csv',
                          encoding='utf-8', thousands=',', skiprows=[1], usecols=[0, 1])
    for tup in df_temp.itertuples():
        df.at[i, tup[1]] = tup[2]
df.to_csv('csvResult/所有国家 15 日确诊人数.csv', index=False)
for i in range(2, 16).__reversed__():
    df.loc[i] -= df.loc[i - 1]
```

```

df.loc[str(i) + ' rate'] = df.loc[i] / df.loc[i - 1]

df = df.T

df['Name'] = np.array(countries)

df['Sum'] = df[df.columns[1:15]].sum(axis=1)

df['Rate Sum'] = df[df.columns[15:29]].mean(axis=1)

df.to_csv('csvResult/所有国家的新增确诊数数据日变化.csv', index=False)

df.sort_values(by='Sum', inplace=True, ascending=False)

df.reset_index(drop=True)

# 作图

# 取出累计确诊最多的 10 个国家

df_res = df[0:10]

fig, ax = plt.subplots()

ax.set_title('12 月 1 日至 15 日每日新增确诊数累计排名前 10 的国家', size=13)

plt.show()

```

图10 关键代码

### 3.2.4 累计确诊人数占国家总人口比例最高的 10 个国家

利用12月15日的数据，分析累计确诊人数占国家总人口比例最高的10个国家，得到如下柱状图：



图11 累计确诊人数占国家总人口比例最高的10个国家

安道尔公国是累计确诊人数占国家总人口比例最高的国家，这是由于其位于欧洲南部，紧邻欧洲疫情中心，且其本身土地面积小（468平方公里）、人口总数少（77321人）。美国这是这10个国家中人口总数最高的国家。

关键代码实现如下：

```
df = pd.read_csv('csvFile/Covid19Data2020-12-15.csv', encoding='utf-8', skiprows=[1], thousands=',',
usecols=[0, 1, 14])

df['Confirmed rate'] = df['Confirmed'] / df['Population']

# 取出并显示累计确诊人数占国家总人口比例最高的 10 个国家

df_res = df[0:10]

plt.tight_layout()

plt.savefig('imgResult/累计确诊人数占国家总人口比例最高的 10 个国家.png')

plt.show()

df_res.to_csv('csvResult/累计确诊人数占国家总人口比例最高的 10 个国家.csv', index=False)
```

图 12 关键代码

### 3.2.5 死亡率最低的 10 个国家

利用12月15日的数据，分析死亡率最低的10个国家，得到如下柱状图：



图13 死亡率最低的10个国家

新加坡是全球新冠疫情死亡率最低的国家，这也是在后续分析应对疫情最好的国家中，该国取得较好成绩的主要原因。



关键代码实现如下：

```
# 取出死亡率最低的 10 个国家
df_res = df[0:10]

df_res = df_res.reset_index(drop=True) # 重置索引
print(df_res)

plt.bar(list(range(0, 50, 5)), df_res['Deceased rate'].to_list(), width=2, alpha=0.5, color='y')
plt.xticks(list(range(0, 50, 5)), labels=df_res['Name'].to_list(), rotation=35)
plt.yticks([0.000, 0.001, 0.002, 0.003, 0.004, 0.005], ['0', '0.1%', '0.2%', '0.3%', '0.4%', '0.5%'])
plt.tick_params(labelsize=9)

for a, b in zip(list(range(0, 50, 5)), df_res['Deceased rate'].to_list()): # 在直方图上显示数字
    plt.text(a, b + 0.000001, '%.2f%%' % (b * 100), ha='center', va='bottom', fontsize=10, color='black')

plt.title('死亡率最低的 10 个国家')
plt.xlabel("国家")
plt.ylabel("死亡率")
```

图 14 关键代码

### 3.2.6 展示各个国家的累计确诊人数的比例

在 15 日数据的基础上，利用饼图绘制展示各个国家的累计确诊人数的比例，饼图显示如下：

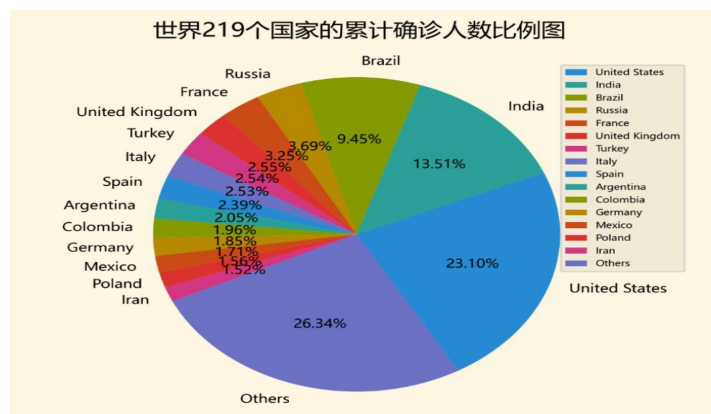


图 15 世界 219 个国家的累计确诊人数比例图

其中，确诊人数小于一百万的国家被归为其它。

关键代码实现如下：

```
"""
用饼图展示各个国家的累计确诊人数的比例（你爬取的所有国家，数据较小的国家 可以合并处理）；
"""
```

```

"""
df = pd.read_csv('csvFile/Covid19Data2020-12-15.csv', encoding='utf-8', skiprows=[1], thousands=',',
usecols=[0, 1])

df.sort_values(by='Confirmed', inplace=True, ascending=False)

df = df.reset_index(drop=True) # 重置索引

# 展示确诊人数大于一百万的国家，其它国家归为其它

df_show = df[df['Confirmed'] > 1000000]
df_other = df[df['Confirmed'] < 1000000]

new = pd.DataFrame({'Name': 'Others', 'Confirmed': df_other['Confirmed'].sum()}, index=[1])
df_show = df_show.append(new, ignore_index=True)

print(df_show)

patches, l_text, p_text = plt.pie(df_show['Confirmed'], startangle=300, labels=df_show['Name'],
                                autopct='%1.2f%%', labeldistance=1.1, textprops={'fontsize': 10,
'color': 'black'})

plt.title('世界 219 个国家的累计确诊人数比例图', y=1.05)

plt.axis('equal')

```

图 16 关键代码

### 3.2.7 展示全球各个国家累计确诊人数的箱型图

通过第 15 日数据分析做出确诊人数箱型图如下：

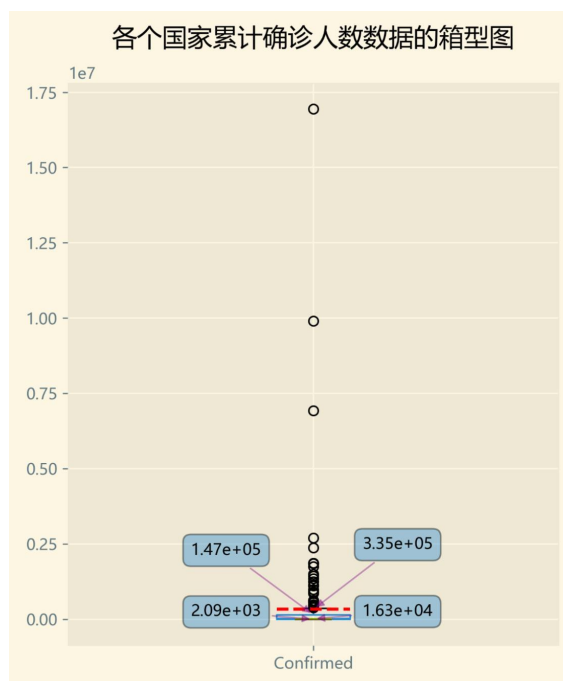


图17 原始箱型图

可以看出，由于异常值过大，箱型图中的图例难以准确展示，且归一化后效果也不好。由于在此图中可以明显看出数据左偏较为严重，因此，删除确诊数据中最高的5条，结果如下：

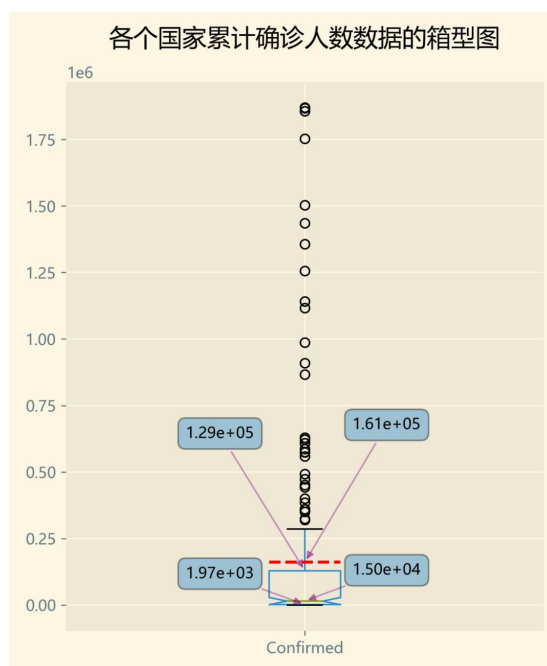


图18 处理后的箱型图

此时箱型图的数据就相对明显了。为了方便观察，使用凹口的形式展现箱线图，均值用红色虚线标注。可以看出下限与下四分位数( $1.97\text{e}+03$ )十分接近（实际上下限为 1），中位数（ $1.50\text{e}+04$ ）和均值( $1.61\text{e}+05$ )偏差较大。这说明数据分布具有较高的偏态，偏态的形态表现为右偏，其实际意义是只有较少部分的国家疫情非常严重，但这些国家的疫情规模也往往远大于其它国家。

关键代码实现如下：

```
# 数据右偏，去除最大的五个异常数据
for i in range(5):
    df = df[df['Confirmed'] != df['Confirmed'].max()]
print(df)
print(df.describe())
# 归一化
# 突出展示均值线
f = df.boxplot(column=['Confirmed'], meanline=True, showmeans=True, vert=True, notch=True,
               return_type='dict', grid=True)
for mean in f['means']:
    mean.set(color='r', linewidth=2)
plt.title('各个国家累计确诊人数数据的箱型图', y=1.05)
```

图 19 关键代码

### 3.2.8 康复率最高的 10 个国家

为了后续分析的便利，额外分析各国新冠疫情存活率情况，得到康复率最高的 10 个国家数据。

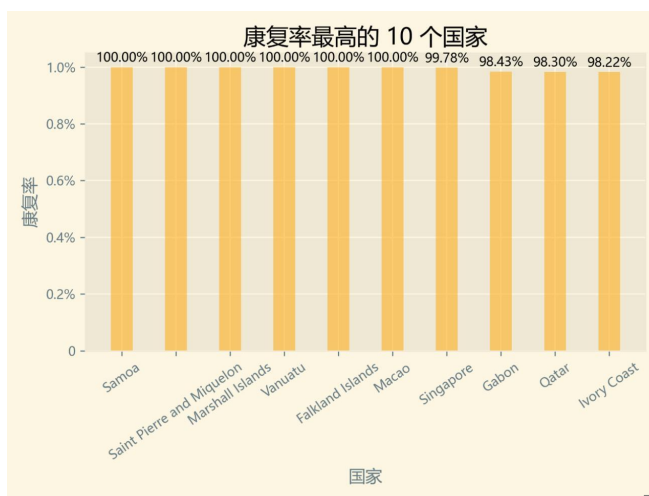


图 20 康复率最高的 10 个国家

可以看出，只有很少部分的国家实现了确诊人数完全康复，并且这些国家都是人口、面积较小的小国家。

关键代码实现如下：

```
"""
康复率(康复人数/确诊人数)最高的 10 个国家:
"""

# 取出死亡率最低的 10 个国家
df_res = df[0:10]

for a, b in zip(list(range(0, 50, 5)), df_res['Recovered rate'].to_list()): # 在直方图上显示数字
    plt.text(a, b + 0.008, '%.2f%%' % (b * 100), ha='center', va='bottom', fontsize=9, color='black')

plt.title('康复率最高的 10 个国家')
plt.xlabel("国家")
plt.ylabel("康复率")
```

图 21 关键代码

### 3.2.9 分析全世界应对新冠疫情最好的 10 个国家

鉴于各个国家人口、医疗水平、检测人数等客观条件不同，故判断应对新冠疫情是否成功的关键因素及其权值如下：

1. 累计确诊人数占国家总人口比例，权值：0.25
2. 死亡率(死亡人数/确诊人数)，权值：0.3
3. 康复率(康复人数/确诊人数)，权值：0.3
4. 15 日内确诊人数日增长率((今日确诊人数-昨日确诊总人数)/昨日确诊总人数)平均值，权值：0.15

权值如此设置是因为在本模型中，新冠肺炎疫情对人的生命健康的影响是首要考虑因素，存活率和康复率是应对疫情是否成功的最关键指标；累计确诊人数是第二重要的因素，因为这表明了某个国家疫情的规模，体现了该国家疫情的严重程度；日增长速率表明了新冠肺炎疫情在某个国家的肆虐速度，一定程度上能体现疫情在该国家的严重程度。但是，应当考虑部分国家由于应对措施得当导致潜在感染者被大量发现的情况，因此该指标的权值最小。

之后，应当设置一个判决值，其计算公式为：

判决值 = (1-确诊率) \* 0.25 + (1-死亡率) \* 0.3 + (1-康复率) \* 0.1（判决值越高的国家，认为其应对新冠疫情做的最好分析）。得到如下表格：

表格1 判决值最优的 10 个国家

	Confirmed	Deceased	Recovered	Increase rate	Judgement
	rate	rate	rate		Value
Singapore	0.00993672	0.000497078	0.997754581	0.000138495	0.996672297
Ivory Coast	0.000813816	0.00612762	0.982170007	0.001242433	0.992422897
Gabon	0.004157353	0.006737247	0.984279756	0.001055578	0.992065078
Ghana	0.001690264	0.006168182	0.976364734	0.001843524	0.990359871
Djibouti	0.005764752	0.010640153	0.980638409	0.000676389	0.98945683
Equatorial Guinea	0.003642538	0.016393443	0.975506268	0.000442679	0.986756811
Uzbekistan	0.00223344	0.008133863	0.963862788	0.002070078	0.985849806
Comoros	0.000715327	0.011146497	0.964968153	0.001972397	0.985671805
Madagascar	0.000627897	0.014726787	0.966168192	0.001011359	0.985123744

### 3.2.10 疫情预测

针对全球累计确诊数，利用前 10 天采集到的数据做后 5 天的预测，并与实际数据进行对比。在该部分，采用了三种预测方法进行预测分析，分别为：

1. 霍尔特(Holt)线性趋势法：水平参数：1，趋势参数：0.2
2. 自回归移动平均模型（ARIMA）：参数  $p$ ， $d$ ， $q$  分别为 2，1，7
3. 滑动窗口时间预测模型：窗口大小 2、3、4

选择霍尔特(Holt)线性趋势法是因为，累计确诊数数据没有季节性，但有递增趋势，该方法可以在无需假设的情况下，准确预测出数据趋势。

自回归移动平均模型（ARIMA）的目标是描述数据中彼此之间的关系，尽管常用来描述数据季节性特征，但同时也能处理具有趋势性的数据预测。

滑动窗口模型则是经典的基于时间序列的预测方法。

利用前 10 日数据作为训练集，后 5 日作为测试集，得到如下分析图：

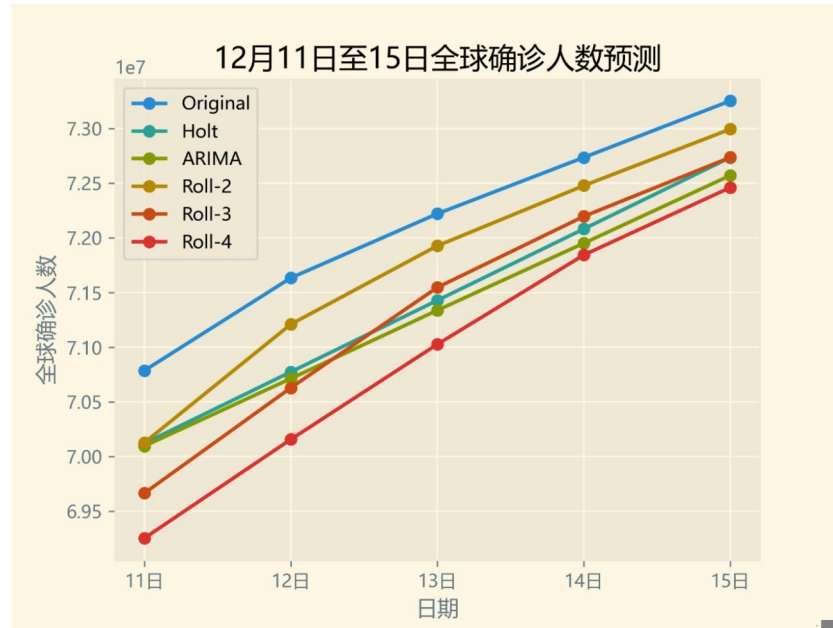


图 22 全球确诊人数预测

预测结果评价采用均方根误差，通过将预测结果和测试集数据进行分析计算，得到如下误差分析表：

表格 2 均方根误差分析表

Method	Value
Holt	707708
ARIMA	797906
Roll-2	408442
Roll-3	809689
Roll-4	1214687

此处没有使用数据归一化的原因在于，由于数据间差距过大，在数据归一化后计算均方误差时将出现较为严重的精度问题，故不予考虑。

由表格 2 及图 20 可知，滑动窗口数为 2 的时间预测方法拥有最好的预测效果，各个方案的详细预测数据如下表所示：

表格 3 详细预测数据

Confirmed	Holt_linear	ARIMA	Roll_2	Roll_3	Roll_4
63649292	63649292	63649292			
64327991	64327991	64327991	63988641.5		

64905069	64905069	64905069	64616530	64294117.33	
65744839	65744839	65744839	65324954	64992633	64656797.75
66294914	66294914	66294914	66019876.5	65648274	65318203.25
66936322	66936322	66936322	66615618	66325358.33	65970286
67493569	67493569	67493569	67214945.5	66908268.33	66617411
68017845	68017845	68017845	67755707	67482578.67	67185662.5
68741600	68741600	68741600	68379722.5	68084338	67797334
69466126	69466126	69466126	69103863	68741857	68429785
70785317	70119936.77	70095029.15	70125721.5	69664347.67	69252722
71633210	70773747.53	70712571.26	71209263.5	70628217.67	70156563.25
72220080	71427558.3	71337257.95	71926645	71546202.33	71026183.25
72734967	72081369.06	71950725.83	72477523.5	72196085.67	71843393.5
73252453	72735179.83	72570695.76	72993710	72735833.33	72460177.5

对于 Holt 和 ARIMA 预测方法来说,预测数据与实际数据存在差异的主要原因在于训练数据过少,并且没有很好地调整参数以适应新冠疫情的增长趋势。

对于基于时间序列的滑动窗口分析来说,其无法很好的应对疫情确诊数据的突发性,即当某些国家的确诊数据突然增大或是减少,该预测方案将不能很好的应对。然而由前文所述,12 月 1 日至 15 日的新冠疫情确诊数据是相对平稳的,因此滑动窗口预测的效果会好于其它两种预测方法。

关键代码实现如下:

```
# 1.原始数据
ax.plot(np.arange(5), df.loc[10:14, 'Confirmed'], marker='o')

# 2.霍尔特(Holt)线性趋势法
df.loc[0:9, 'Holt_linear'] = df.loc[0:9, 'Confirmed']
fit = Holt(np.asarray(df.loc[0:9, 'Confirmed'])).fit(smoothing_level=1, smoothing_trend=0.2)
df.loc[10:14, 'Holt_linear'] = fit.forecast(5)
ax.plot(np.arange(5), df.loc[10:14, 'Holt_linear'], marker='o')
df_error.at[0, 'Value'] = sqrt(mean_squared_error(df.loc[10:14, 'Confirmed'], df.loc[10:14, 'Holt_linear']))

# 3.自回归移动平均模型 (ARIMA)
df.loc[0:9, 'ARIMA'] = df.loc[0:9, 'Confirmed']
fit1 = sm.tsa.statespace.SARIMAX(df.loc[0:9, 'Confirmed'], order=(2, 1, 7)).fit()
df.loc[10:14, 'ARIMA'] = fit1.predict(start=10, end=14, dynamic=True)
```



```
ax.plot(np.arange(5), df.loc[10:14, 'ARIMA'], marker='o')

df_error.at[1, 'Value'] = sqrt(mean_squared_error(df.loc[10:14, 'Confirmed'], df.loc[10:14, 'ARIMA']))

# 4.生成滑动窗口为 2 的预测值

df['Roll_2'] = df['Confirmed'].rolling(window=2, center=False).mean()

ax.plot(np.arange(5), df.loc[10:14, 'Roll_2'], marker='o')

df_error.at[2, 'Value'] = sqrt(mean_squared_error(df.loc[10:14, 'Confirmed'], df.loc[10:14, 'Roll_2']))

# 5.生成滑动窗口为 3 的预测值

df['Roll_3'] = df['Confirmed'].rolling(window=3, center=False).mean()

ax.plot(np.arange(5), df.loc[10:14, 'Roll_3'], marker='o')

df_error.at[3, 'Value'] = sqrt(mean_squared_error(df.loc[10:14, 'Confirmed'], df.loc[10:14, 'Roll_3']))

# 6.生成滑动窗口为 4 的预测值

df['Roll_4'] = df['Confirmed'].rolling(window=4, center=False).mean()

ax.plot(np.arange(5), df.loc[10:14, 'Roll_4'], marker='o')

df_error.at[4, 'Value'] = sqrt(mean_squared_error(df.loc[10:14, 'Confirmed'], df.loc[10:14, 'Roll_4']))
```

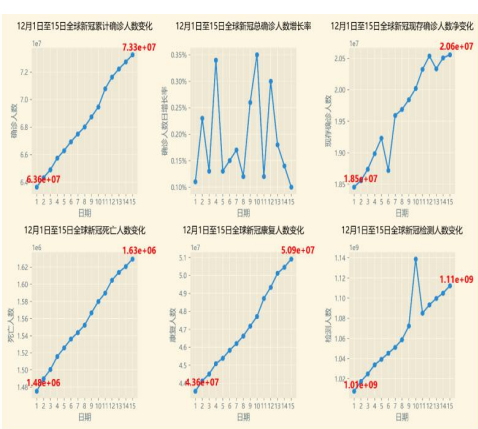
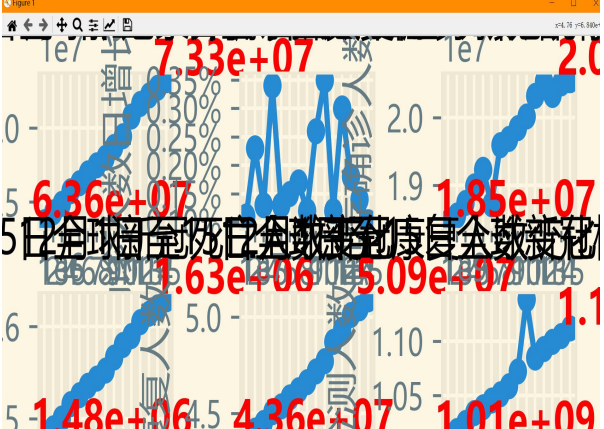
图 23 关键代码

## 四、 设计日志

### 4.1 界面无法正常显示

Python 图形化界面无法正常显示，字体过大、图像显示错位等问题；

表格 4 问题 1

1、期望显示效果：	2、pycharm 实际运行效果如下：
	

解决办法:

第一步:

1.导入正确版本的matplotlib库

2.代码最后调用matplotlibku中pyplot.show()方法

图 24 解决办法

第二步: 调整电脑 dpi, 缩小电脑缩放尺寸;

最后能够正常显示。

## 4.2 无法导入相应包

出现 ImportError: No module named pandas;

```
C:\Users\Administrator\Desktop\Covid19Spider-main>python part-A.py
Traceback (most recent call last):
  File "part-A.py", line 6, in <module>
    import pandas as pd
ModuleNotFoundError: No module named 'pandas'
```

图 25 问题

解决办法: 需要在导入包之前先激活虚拟环境, 之后能正常导入包。

```
C:\Users\Administrator\Desktop\Covid19Spider-main>activate tensorflow
(tensorflow) C:\Users\Administrator\Desktop\Covid19Spider-main>python part-A.py
1          2  ...          14
15
Name      TOTAL      TOTAL      TOTAL      TOT
```

图 26 解决办法

## 五、 个人小结

曹蓉:

本次实验选题比较有难度, 从确定选题到开展实验之间遇到了一系列的问题, 花费了我不少时间。

首先非常感谢我的指导老师刘老师, 他在课上用图形与实例结合的方式生动形象地向我展示了进行数据分析和实施的过程与方法, 他给我指明了如何进行实

验的方向；

此外还要感谢我的队友，谢谢他的理解与支持，我们小组最终能取得这样的成绩非常不容易；

最后感谢我自己，从一个不会写 python 代码、不了解算法的小白，到愿意接连好几天啃 python 基础语法和高级算法，到 B 站上跟着教学视频了解 Pycharm、python 图形库的使用和文件的处理，查阅大量文献与资料，并最终完成了此次 python 项目，总体来说此次项目对我个人提升很大。

虽然项目的实施取得了一定进展，但是仍然有不足的地方，比如影响疫情的因素是多种多样的，如何弱化影响因素的影响以及如何选择影响因素，这些都是之后需要优化的地方。

杨孟衡：

通过完成这次 python 课程设计，培养了我的实际分析问题和动手能力，使我更加充分的掌握了课本上所学不到的知识，并能够应用于实践当中。本次 python 设计课程，让我觉得学术遥远，自己才疏学浅。应用软件的研究何其庞杂，何其精妙，这次设计其实只能是涉其皮毛，距离理想之境还有很长的路。

在程序设计的过程中，投入了许多的精力与时间。主要是在分析网页结构与图形化调整上的时间花费的比较多，中间也改了很多次设计，不过最后我们小组两个人都顶住了压力，积极努力，最终解决了设计难点。

回顾起本学期的 python 课程，学完第三方库之后发现 python 第三方库函数实现的功能非常多，网上学习了一下来应用到项目中。从开始设计到顺利实现设计，从理论到实践，在整个学习的日子里，可以学到很多很多的东西，不仅可以巩固了以前所学过的知识，而且还学到了很多在书本上所没有学到过的知识。在设计中遇到了很多难以解决的问题，最后在自己的辛勤努力下迎刃而解，这种收获是非常难得的，值得珍惜。

本次 python 课程设计能够顺利完成，使自己熟练掌握了许多能力，这些都多亏了老师在课堂上不倦地指导和教诲以及同组队友的无私帮助，感谢老师和我的队友！

## 六、 参考文献

- [1] 曾长清等著.Python 编程——乐学程序设计与数据处理.电子工业出版社.2020.11
- [2] 朝乐门 著.Python 编程从数据分析到数据科学.电子工业出版社.2019.1
- [3] [美] 阿曼多·凡丹戈 (Armando Fandango) 著, 韩波 译. Python 数据分析(第 2 版). 人民邮电出版社.2018.6
- [4] 嵩天, 礼欣, 黄天羽 著.Python 语言程序设计基础 (第 2 版). 高等教育出版社.2017.2