



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 世界新冠疫情数据分析

班 级 金融201班

学 号 0204767

姓 名 陈思博

任课教师 肖 泉

开课学期: 2020 至 2021 学年 第 2 学期

完成时间: 2021 年 6 月 29 日

《新冠疫情数据分析》数据分析报告

目 录

1 概述	1
2 数据描述	1
3 数据分析内容	5
(1) 取全球感染人数前五的国家进行统计，并可视化分析.....	5
(2) 人口总数与人口密度对于疫情的爆发程度影响进行分析。 5	
(3) 防疫公共措施（洗手措施）是否能有效减少感染人数的增加 分析。	5
(4) 国家的重症患者数量是否与糖尿病患者概率的高低有关分析	6
4 数据分析图表	6
5 数据分析结果	10
(1) 结果 1:	10
(2) 结果 2:	10
(3) 结果 3:	11
(4) 结果 4:	11
6 总结	11
附录-数据分析代码	11

1 概述

本数据背景是基于 2019 年至今爆发的全球性疫情而建立的，本数据分析项目取 2020 年初期和 2021 年初期数据进行分析，旨在分析全球各地区爆发疫情的严重程度与该区人数比例的关系、疫情严重程度与国家实力的关联：通过对这些问题的分析与可视化可以更清楚的看出全球各个国家针对疫情的具体措施效果，也可以更好的分析“人类命运共同体”的建立是否有利于人类的发展；总的来说，基于本数据的分析后，可以更直观的看出疫情对于不同国家的冲击，也可以看出不同国家对疫情的应对措施效果，一些客观因素对于疫情的影响和其相关性强弱。

2 数据描述

本数据是从“阿里云天池”的网络数据平台后取得，包含了全球各个洲与各个国家的疫情增长情况字段名为 total_cases 和 new_cases，死亡情况字段 total_deaths 和 new_deaths，新冠肺炎再发性概率 reproduction_rate，重症病人和住院病人字段 icu_patients 和 hosp_patients，接种疫苗的人数字段 total_vaccinations 以及人口字段 population、人均 GDP 字段 gdp_per_capita 等重要字段，该数据的特点是数据量大，数据分析的特征有很多，但是，数据里面有很多 Nan 值，即有缺失值和未统计的值，所以，对于本次数据分析项目所需要或有所扰乱的值进行了填充与拆分合并，旨在将数据的操作误差降到最小。

各字段是否含有空值情况：

iso_code	False
continent	True
location	False
date	False
total_cases	True
new_cases	True
new_cases_smoothed	True
total_deaths	True
new_deaths	True
new_deaths_smoothed	True
total_cases_per_million	True
new_cases_per_million	True
new_cases_smoothed_per_million	True
total_deaths_per_million	True
new_deaths_per_million	True
new_deaths_smoothed_per_million	True
reproduction_rate	True
icu_patients	True
icu_patients_per_million	True
hosp_patients	True
hosp_patients_per_million	True
weekly_icu_admissions	True
weekly_icu_admissions_per_million	True
weekly_hosp_admissions	True
weekly_hosp_admissions_per_million	True
new_tests	True
total_tests	True

首先，对该数据进行总体的缺失值分析：

total_tests_per_thousand	True
new_tests_per_thousand	True
new_tests_smoothed	True
new_tests_smoothed_per_thousand	True
positive_rate	True
tests_per_case	True
tests_units	True
total_vaccinations	True
people_vaccinated	True
people_fully_vaccinated	True
new_vaccinations	True
new_vaccinations_smoothed	True
total_vaccinations_per_hundred	True
people_vaccinated_per_hundred	True
people_fully_vaccinated_per_hundred	True
new_vaccinations_smoothed_per_million	True
stringency_index	True
population	True
population_density	True
median_age	True
aged_65_older	True
aged_70_older	True
gdp_per_capita	True
extreme_poverty	True
cardiovasc_death_rate	True
diabetes_prevalence	True
female_smokers	True

male_smokers	True
handwashing_facilities	True
hospital_beds_per_thousand	True
life_expectancy	True
human_development_index	True

发现数据的空值有很多，但是大多都是数值类型，所以大部分的数值因为是 0 所以统计的表格中是空值，所以可以利用拆分数据，将有用的数据 Nan 值进行填充 0，合并数据后，将数据中没有统计国家名字的丢失数据进行剔除，即是将 continent 的数据段中的空值剔除（其具有的人类发展指数字段也为 0）并对区域内无疫情区域进行剔除，防止海外殖民地和岛屿及一些未统计的地区对分析造成干扰，后续的分析中会利用该手法剔除掉没有统计地区

各字段是否含有空值情况：

iso_code	False
location	False
date	False
total_cases	False
new_cases	False
new_cases_smoothed	False
total_deaths	False
new_deaths	False
new_deaths_smoothed	False
total_cases_per_million	False
new_cases_per_million	False
new_cases_smoothed_per_million	False
total_deaths_per_million	False
new_deaths_per_million	False
new_deaths_smoothed_per_million	False
reproduction_rate	False
icu_patients	False
icu_patients_per_million	False
hosp_patients	False
hosp_patients_per_million	False
weekly_icu_admissions	False
weekly_icu_admissions_per_million	False
weekly_hosp_admissions	False
weekly_hosp_admissions_per_million	False

的值，进行更有效地统计分析：

new_tests	False
total_tests	False
total_tests_per_thousand	False
new_tests_per_thousand	False
new_tests_smoothed	False
new_tests_smoothed_per_thousand	False
positive_rate	False
tests_per_case	False
tests_units	False
total_vaccinations	False
people_vaccinated	False
people_fully_vaccinated	False
new_vaccinations	False
new_vaccinations_smoothed	False
total_vaccinations_per_hundred	False
people_vaccinated_per_hundred	False
people_fully_vaccinated_per_hundred	False
new_vaccinations_smoothed_per_million	False
stringency_index	False
population	False
population_density	False
median_age	False
aged_65_older	False
aged_70_older	False
gdp_per_capita	False
extreme_poverty	False
cardiovasc_death_rate	False

```

diabetes_prevalence      False
female_smokers            False
male_smokers              False
handwashing_facilities   False
hospital_beds_per_thousand False
life_expectancy          False
human_development_index  False
continent                False
dtype: bool

```

数据清洗完成，已无空值。

利用 data.describe() 对数据填充后的异常值进行分析：

	total_cases	new_cases	new_cases_smoothed	total_deaths \
count	6.490600e+04	64906.000000	64906.000000	64906.000000
mean	1.871792e+05	1669.064832	1652.831680	4946.672696
std	1.054969e+06	9241.501067	9022.530983	22414.218490
min	1.000000e+00	-46076.000000	-1121.714000	0.000000
25%	5.690000e+02	1.000000	4.429000	9.000000
50%	5.965500e+03	47.000000	54.429000	107.000000
75%	6.093625e+04	520.000000	532.286000	1135.000000
max	2.769416e+07	300282.000000	249836.143000	486325.000000

	new_deaths	new_deaths_smoothed	total_cases_per_million \
count	64906.000000	64906.000000	64906.000000
mean	37.097341	36.643470	6529.272274
std	173.466718	160.466709	13165.227534
min	-1918.000000	-232.143000	0.001000
25%	0.000000	0.000000	120.992000
50%	0.000000	0.714000	935.977000
75%	8.000000	8.857000	5840.707000
max	5443.000000	3364.143000	136387.756000

	new_cases_per_million	new_cases_smoothed_per_million \
count	65721.000000	65721.000000
mean	62.247913	61.421221
std	167.014487	138.845892
min	-2153.437000	-276.825000
25%	0.045000	0.713000
50%	4.997000	6.422000
75%	48.734000	53.353000
max	8652.658000	2648.773000

	total_deaths_per_million	new_deaths_per_million \
count	65721.000000	65721.000000
mean	133.348425	1.144724
std	270.434776	3.685057
min	0.000000	-76.445000
25%	1.262000	0.000000
50%	16.213000	0.000000
75%	103.779000	0.686000
max	2121.516000	218.329000

	new_deaths_smoothed_per_million	reproduction_rate	icu_patients	\
count	65721.000000	65721.000000	65721.000000	
mean	1.129453	0.797954	111.381902	
std	2.771287	0.535573	986.136606	
min	-10.921000	0.000000	0.000000	
25%	0.000000	0.260000	0.000000	
50%	0.078000	0.960000	0.000000	
75%	0.822000	1.140000	0.000000	
max	63.140000	6.720000	23912.000000	

	icu_patients_per_million	hosp_patients	hosp_patients_per_million	\
count	65721.000000	65721.000000	65721.000000	
mean	2.394313	668.816817	19.301220	
std	10.619569	5341.524961	84.176995	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	
max	127.183000	132474.000000	1042.535000	

	new_tests	total_tests	total_tests_per_thousand	\
count	6.572100e+04	6.572100e+04	65721.000000	
mean	1.758488e+04	1.961533e+06	68.913942	
std	1.022851e+05	1.336339e+07	216.656121	
min	-2.391720e+05	0.000000e+00	0.000000	
25%	0.000000e+00	0.000000e+00	0.000000	
50%	0.000000e+00	0.000000e+00	0.000000	
75%	4.289000e+03	3.825050e+05	31.010000	
max	2.945871e+06	3.124389e+08	3153.678000	

	new_tests_per_thousand	new_tests_smoothed	\
count	65721.000000	6.572100e+04	
mean	0.654237	1.923293e+04	
std	2.551217	1.003653e+05	
min	-23.010000	0.000000e+00	
25%	0.000000	0.000000e+00	
50%	0.000000	2.440000e+02	
75%	0.401000	6.140000e+03	
max	327.086000	1.851168e+06	

	total_vaccinations	people_vaccinated	people_fully_vaccinated	\
count	6.572100e+04	6.572100e+04	6.572100e+04	
mean	3.837051e+04	2.542124e+04	5.245242e+03	
std	8.067253e+05	6.006677e+05	1.587562e+05	
min	0.000000e+00	0.000000e+00	0.000000e+00	
25%	0.000000e+00	0.000000e+00	0.000000e+00	
50%	0.000000e+00	0.000000e+00	0.000000e+00	
75%	0.000000e+00	0.000000e+00	0.000000e+00	
max	5.288436e+07	3.829227e+07	1.407744e+07	

	new_vaccinations	new_vaccinations_smoothed	\
count	6.572100e+04	6.572100e+04	
mean	1.704652e+03	2.412249e+03	
std	3.448927e+04	3.873923e+04	
min	0.000000e+00	0.000000e+00	
25%	0.000000e+00	0.000000e+00	
50%	0.000000e+00	0.000000e+00	
75%	0.000000e+00	0.000000e+00	
max	2.242472e+06	1.916190e+06	

	handwashing_facilities	hospital_beds_per_thousand	life_expectancy
count	65721.000000	65721.000000	65721.000000
mean	24.484932	2.688292	72.732011
std	33.741925	2.524667	9.316152
min	0.000000	0.000000	0.000000
25%	0.000000	0.800000	67.270000
50%	0.000000	2.100000	74.620000
75%	47.964000	3.800000	78.730000
max	98.999000	13.800000	86.750000

	human_development_index
count	65721.000000
mean	0.702201
std	0.200450
min	0.000000
25%	0.583000
50%	0.743000
75%	0.845000
max	0.957000

进行分析后发现并无异常值。

3 数据分析内容

(1) 取全球感染人数前五的国家进行统计，并可视化分析

人类发展指数数据：“人类发展指数（HDI——Human Development Index）是由联合国开发计划署（UNDP）在《1990 年人文发展报告》中提出的，用以衡量联合国各成员国经济社会发展水平的指标，是对传统的 GNP 指标挑战的结果。人类发展指数从动态上对人类发展状况进行了反映，揭示了一个国家的优先发展项，为世界各国尤其是发展中国家制定发展政策提供了一定依据，从而有助于挖掘一国经济发展的潜力。”且人类发展指数越高，证明该国的综合实力越高。借此以全球感染人数前五的国家，结合人类发展指数进行评估，旨在分析人类发展指数对疫情的控制有效是否相关。分析有利于更好的看出不同 HDI 水平的国家是否用自身优势去抑制疫情，措施的影响是否大于 HDI 的影响。

(2) 人口总数与人口密度对于疫情的爆发程度影响进行分析。

人口密度：“人口密度是单位土地面积上的人口数量。通常使用的计量单位有两种：人/平方公里;人/公顷。它是衡量一个国家或地区人口分布状况的重要指标。计算人口密度的土地面积是指领土范围内的陆地面积和内陆水域，不包括领海。由于人口密度指标是假定人口均匀分布在它所涉及的一定地域内，因此，人口密度计算的范围愈小，就愈能如实地反映人口分布的情况;范围愈大则只能概括地揭示人口分布的大势。”对疫情的感染人数与人口做相关分析，进行可视化数据分析，分析有利于更直观的看到人口基数或是人口密度不同的国家如何采取有效的措施去抑制不利因素可能会带来的疫情加重影响，给世界其他国家提供了抗疫“参考模板”。

(3) 防疫公共措施（洗手措施）是否能有效减少感染人数的增加分析。

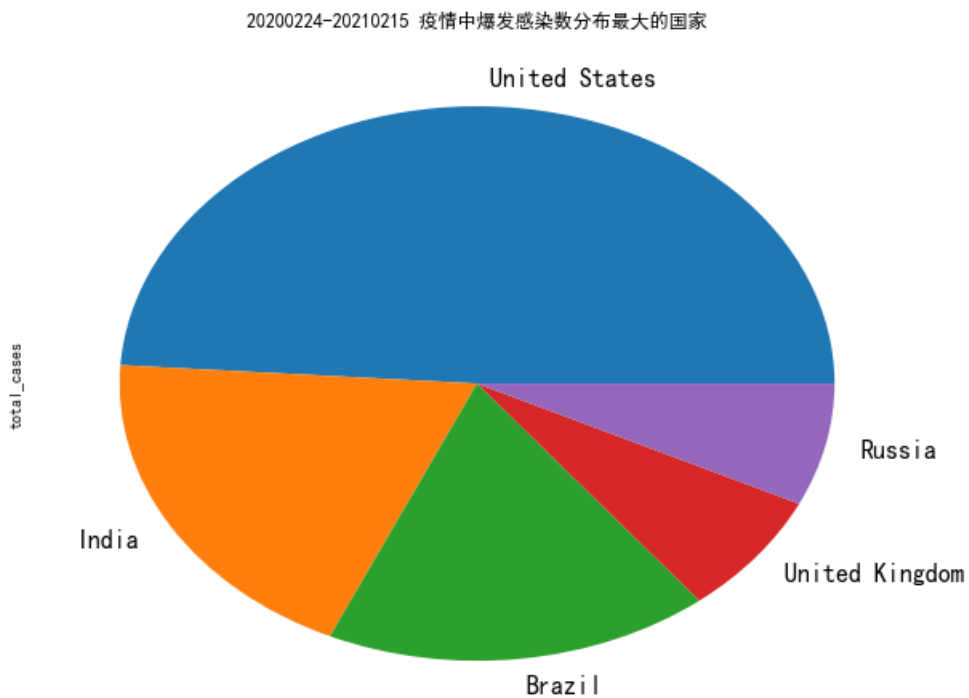
防疫期间的国家公共设施对于疫情的阻隔在大众看来十分有效，因为它阻断了手上的病毒进入人体，造成感染，但是，部分没有公共设施（洗手措施）的国家是否会爆发更严重的疫情；本分析旨在去探索防疫公共措施的建立多少是否与感染人数的多少有所关联。有利于

各国察情况是否加大这类防疫措施。

(4) 国家的重症患者数量是否与糖尿病患者概率的高低有关分析

本分析进行新冠重症患者的数据整合与糖尿病患者概率的数据整合,进行国家的基础病(糖尿病)是否在新冠的影响下造成很大双重影响;本分析旨在为各国的基础病或是并发症高的国家提供关联性的比较,在疫情的影响下,国家患基础病(糖尿病)的概率高的国家是否在患新冠时候的重症患者会增多,有利于国家根据该病做出特殊人群分离,特殊人群在新冠期间的特殊防范,将人员损失降到最低。

4 数据分析图表



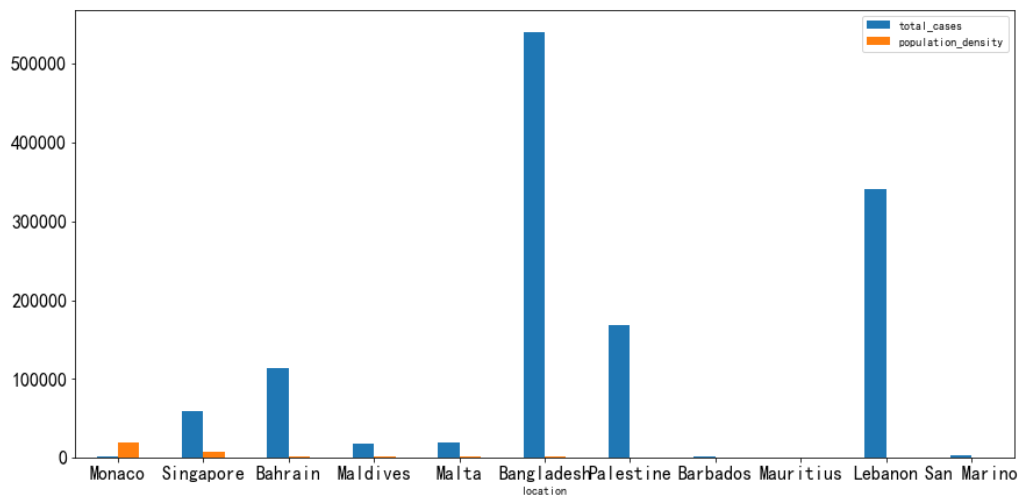
(1)

1-1

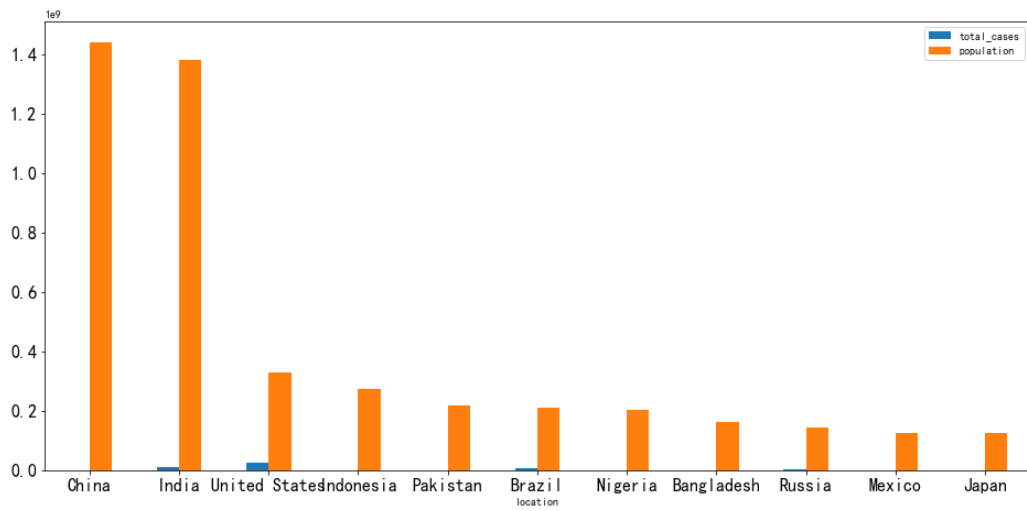
```
United States 0.9259999999999999
India 0.645
Brazil 0.765
United Kingdom 0.932
Russia 0.8240000000000001
```

1-2

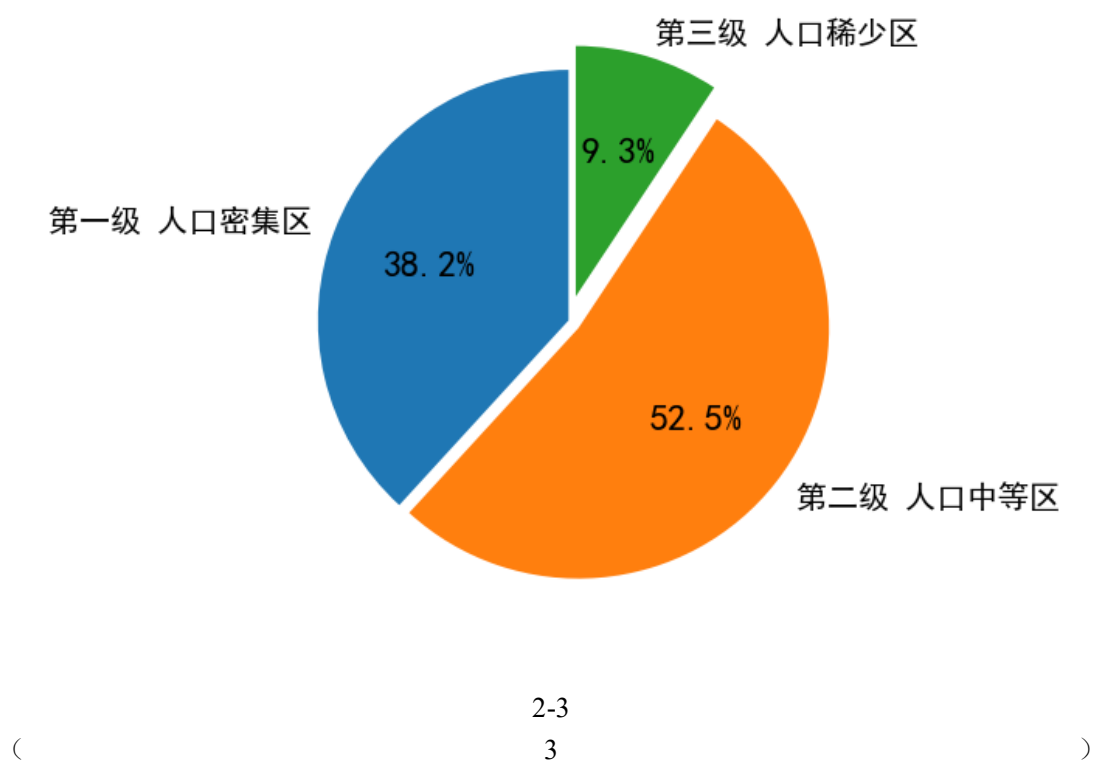
2

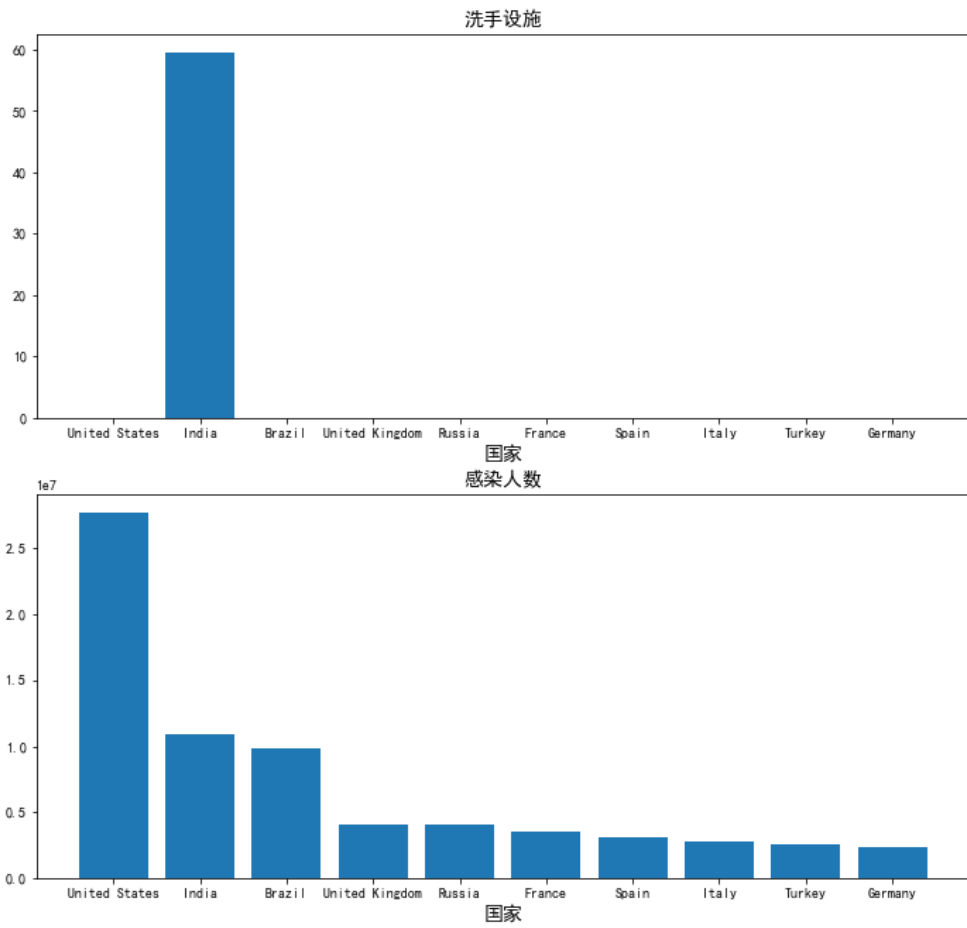


2-1

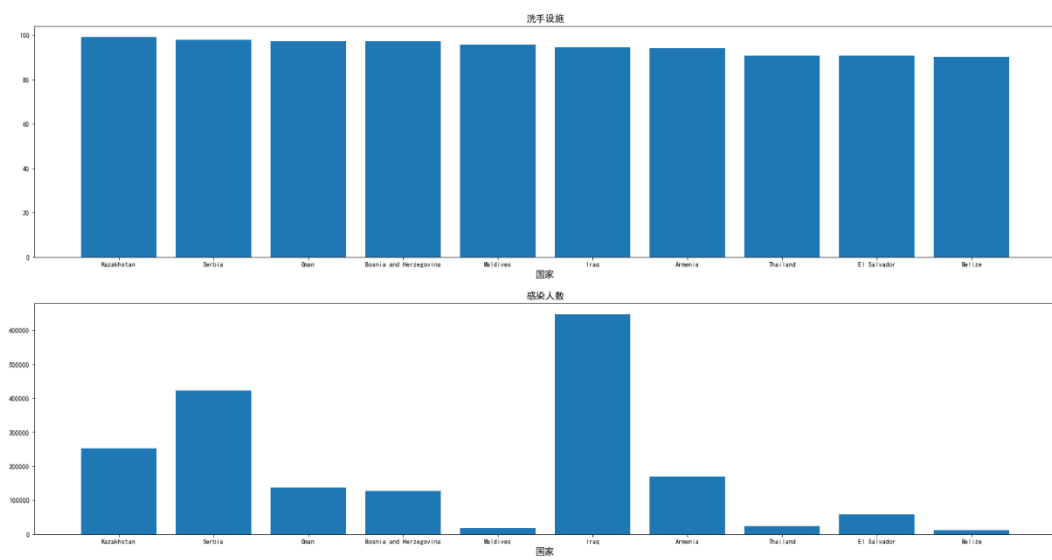


2-2

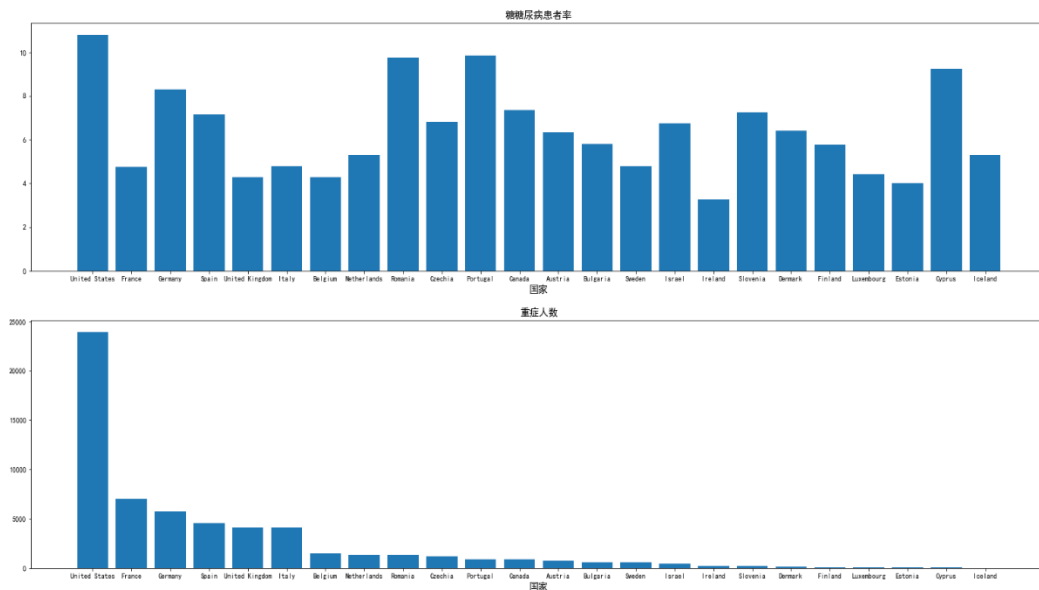




3-1



3-2



5 数据分析结果

(1) 结果 1:

由图 1-1 与图 1-2 可看出美国的人类发展指数 (HDI) 很高, 达到了 0.926 和 0.932, 而印度和巴西的 HDI 并没有美国和英国的高, 但是美国的疫情感染人数是基于本数据下最多的国家, 据最新数据显示印度的感染人数位居全球第二, 而美国依旧高据第一; 而英国虽然疫情感染人数与美国相比较较少, 但是依旧占据很多。综上所述, 高的人类发展指数并不一定能带来更好的疫情控制, 例如美国和英国; 但是 HDI 低的国家更有可能受到疫情的冲击; 这不仅令我们可以看到未来国家在面对自然灾害和各种环境变化时所作出的决策对于国家有很大的影响, 即使时高水平发达国家, 仍然在决策失误的情况下会受到很大的冲击, 在发达国家, 国家决策的因素的影响程度是大于国家是否发达因素的影响程度的, 所以, 高 HDI 的国家不应该懈怠自己的防疫措施, 更应该加强自己的防疫措施, 避免出现人员感染, 贯彻“人类命运共同体”这一终极理想。

(2) 结果 2:

由图 2-1 可知, 人口密度对于疫情爆发程度的有一定的影响, 比如孟加拉国的人口密度排在前面, 他们国家的感染人数也较多, 但是影响并不一定会体现, 比如第一个摩纳哥, 人口密度很大, 但是其感染人数却是很少的, 这就表明了人口密度的不足以影响影响大规模疫情爆发; 由图 2-2 可知, 除了中国外, 人口总数较多的国家感染人数是有一定比列的增加了, 但是相对增加不够明显, 其中有部分还降低了, 但大体上, 人数总数多的国家感染疫情后的爆发程度加大的几率是有所增加的, 例如图中的巴西与美国; 由图 2-3 可以清晰的知道人口稀少区感染风险较少, 但是人口中等区的感染人数超过了人口密集区, 这表明着人口相对分布均匀的国家政策措施有较大的失误, 导致了部分感染的严重。综上所述, 在一定程度下, 人口密度和人口总数的确能在一定程度上影响感染人数的增加, 但是高的人口密度与人口总数的国家在有效的防疫措施下能使其影响减少到最小。

（3）结果 3:

由图 3-1 可知，公共防疫措施（洗手措施）在排名靠前的“自由民主”国家并不有用，体现为美国的公共洗手措施很小几乎为 0，其感染率很高，印度的公共洗手措施很高，但是其感染率依旧很高，这表明在崇尚“自由民主”的国家中，人民对于公共洗手措施并没有用的很好。而事实证明洗手的确可以降低新冠的增加率，所以，数据告诉了我们，防疫措施中，全民自觉遵守的因素强弱程度会比国家建立多少的因素强弱程度更有力影响疫情感染人数。由图 3-2 可知，相较于公共设施的建立少的国家，洗手措施的建立的确可以降低新冠的发生率。综上所述，公共卫生在部分国家用处很大，有效的抑制了疫情的扩散，但是这对公民的自觉性要求很高，故公共卫生的设施建立对于公民自觉度高的国家有很大的影响，但对于“自由民主”类的国家来说并不适用。

（4）结果 4:

由图 4 可知，患糖尿病的概率高低在一定程度上会影响本国患新冠患者的重症率，例如美国的患者重症度很高，其具有的糖尿病患者概率也是很高的，根据统计后的有效数据来看，一个国家具有较高的糖尿病患病率对新冠患者变为重症患者有一定的影响，但在部分感染不严重的国家其感染基数小，所以与糖尿病患者概率的联系并不强。综上所述，糖尿病患病概率的确会影响该国的重症患者数量，并且在大规模疫情入侵下会更加明显，因为感染的人中所患糖尿病概率会大很多，所以，一国在进行防疫措施实施的时候，应该对这类人群采取更有效、更强的防疫隔离措施，以隔绝他们被感染的可能性，因为他们的感染可能造成的是人员损失的后果，对国家造成严重的后果。

6 总结

本次分析采取了拆分合并、重组数据等有效手段去解决数据中出现的不同误差值，有效的规避了大型误差的产生，尽管数据的不全造成了数据与实际的偏差，但是经过后续的再拆分重组，得到的有效数据很好的去证明了本次分析的意义所在；本数据分析共有四个内容，四个内容涵盖了有特殊到一般的规律，即有特殊病种到一般基础病，由特殊防疫设施到一般公共卫生设施等，综合分析了基于世界疫情大爆发的背景下的怎么去防范疫情更有效，而且分析了一些人类公认的指数是否与疫情的防范得当有关联，有利于促进人们可以更好的对“人类命运共同体”这一终极理想有所重视。

本次分析让我了解了 python 数据分析的大部分基础知识，让我知道了 python 可以进行有效的数据分析项目，得出精确的可视化结果；本次数据的代码书写也令我对 python 中基础代码的掌握程度愈发熟练，且深知自己在部分数据面前还是无从下手的浅陋，故在后续的学习中，会更好、更深入的学习 python 这一门科学技术，让它带给我的不仅仅只是片面的知识，而是一些可以辅助我去进行更简便的工作或其他学习的知识应用。

附录-数据分析代码

```
import tushare as ts
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
ts.set_token('e9e6f0d2956bcef3713b48b4aca9b899713f5bb3403988d3c7f6ed90')
```

```
pro=ts.pro_api()
data=pd.read_csv('owid-covid-data.csv')

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
print('各字段是否含有空值情况: \n', data.isna().any())
print(data.info())
print(data.describe())

#设置字体
mpl.rcParams['font.sans-serif'] = 'SimHei'
mpl.rcParams['axes.unicode_minus'] = False

#进行数据清洗 并保存清洗后的数据
df=data['continent']
data1=data.drop('continent',axis=1)
data=data1.fillna(0)
df1=pd.concat([data,df],axis=1)

df3=df1.dropna(axis='index',how='any',subset=['continent'])
df3=df3.query('total_cases>0')
df3.to_excel('数据集.xlsx')

#进行检查数据清洗效果
print('各字段是否含有空值情况: \n', df3.isna().any())
print(df3.info())
print(df3.describe())

#（1）全球前五的国家——进行画图分析
group_1=df3.groupby('location').max()
group_2=group_1.sort_values(by='total_cases',ascending=False)

group_2_1=group_2.head(5)

group_2_1.total_cases.plot(kind='pie',title='20200224-20210215 疫情中爆发感染数分布最大的国家',
                             figsize=(10,8),fontsize=16)

#人类发展指数选取
group_2.reset_index(inplace=True)
df2=group_2['location']
df4=df2.head(5)
```

```
group_3=group_2.groupby('location')['human_development_index'].max()
lst1=df4.tolist()
```

```
for x in lst1:
    d=group_3.loc[x]
    print(x,d)
```

#(2)人口密度和人口总数

```
group_4=group_1.sort_values(by='population_density',ascending=False)
group_4.reset_index(inplace=True)
group_5=group_4.iloc[0:11]
df5=group_5[['total_cases','population_density','location']]
df5=df5.set_index('location')
df5.plot(kind='bar',rot=0,fontsize=16)
```

```
group_6=group_1.sort_values(by='population',ascending=False)
group_6.reset_index(inplace=True)
group_7=group_6.iloc[0:11]
df6=group_7[['total_cases','population','location']]
df6=df6.set_index('location')
df6.plot(kind='bar',rot=0,fontsize=16)
```

#不同人口密集区的感染人数总和

```
group_1.reset_index(inplace=True)
df7=group_1.query('population_density > 100')
df8=group_1.query('population_density > 25 and population_density <= 100')
df9=group_1.query('population_density > 1 and population_density <= 25')
df7=df7['location']
df8=df8['location']
df9=df9['location']
```

```
df7_lst=df7.tolist()
df8_lst=df8.tolist()
df9_lst=df9.tolist()
```

```
group_1=group_1.set_index('location')
group_8=group_1['total_cases']
sum_1=0
sum_2=0
sum_3=0
for i in df7_lst:
    sum_1 += group_8.loc[i]
```

```
for i in df8_lst:
```



```

sum_2 += group_8.loc[i]

for i in df9_lst:
    sum_3 += group_8.loc[i]

l=['第一级 人口密集区','第二级 人口中等区','第三级 人口稀少区']
lst2=[sum_1,sum_2,sum_3]
explode=(0,0.05,0.1)
patches,ltext,ptext
=plt.pie(lst2,explode=explode,labels=l,autopct='%1f%%',shadow=False,startangle=90)
for x in ltext:
    x.set_size(20)

for x in ptext:
    x.set_size(24)
plt.axis('equal')

#(3)公共卫生设施的作用（洗手设施）分析
#将数据进行拆分合并最后再拆分，得到一个只包含本数据分析项目的两个关键字段 Series
df10=df3[['handwashing_facilities','location']]
df10=df10.set_index('location')
group_9=df10.groupby('location').max()
df11=group_9.sort_values(by='handwashing_facilities',ascending=False)

df12=df3[['total_cases','location']]
df12=df12.set_index('location')
df12=df12.groupby('location').max()
df13=pd.concat([df11,df12],axis=1)
df13=df13.sort_values(by='total_cases',ascending=False)
'''
进行关键字段名字代换，输出本分析图 2
'''

df13.reset_index(inplace=True)
df13=df13.rename(columns={'index':'location'})
df13=df13.head(10)

fig=plt.figure()
ax1=fig.add_subplot(211)
ax2=fig.add_subplot(212)
ax1.bar(x=df13['location'],height=df13['handwashing_facilities'])

ax2.bar(x=df13['location'],height=df13['total_cases'])

```

```

ax1.set_xlabel('国家',fontsize=14)
ax2.set_xlabel('国家',fontsize=14)
ax1.set_title('洗手设施',fontsize=14)
ax2.set_title('感染人数',fontsize=14)

#（4）国家的重症患者数量是否与糖尿病患者概率的高低有关分析
#取每个国家在本时期内的最大重症数
df14=df3['icu_patients']
df15=df3['location']
df16=pd.concat([df15,df14],axis=1)
df16=df16.set_index('location')
df16=df16.query('icu_patients > 0 ')
df17=df16.groupby('location')['icu_patients'].max()

#取糖尿病患者概率
df18=df3['diabetes_prevalence']
df19=df3['location']
df20=pd.concat([df18,df19],axis=1)
df20=df20.set_index('location')
df20=df20.query('diabetes_prevalence > 0 ')
df21=df20.groupby('location')['diabetes_prevalence'].mean()
#取与有重症患者的国家数据进行分析
df22=pd.concat([df17,df21],axis=1).dropna()
df23=df22.sort_values(by='icu_patients',ascending=False)

df23.reset_index(inplace=True)
df23=df23.rename(columns={'index':'location'})

fig=plt.figure()
ax3=fig.add_subplot(211)
ax4=fig.add_subplot(212)
ax3.bar(x=df23['location'],height=df23['diabetes_prevalence'])

ax4.bar(x=df23['location'],height=df23['icu_patients'])
ax3.set_xlabel('国家',fontsize=14)
ax4.set_xlabel('国家',fontsize=14)
ax3.set_title('糖尿病患者率',fontsize=14)
ax4.set_title('重症人数',fontsize=14)
'''
注意：代码使用在画图时需进行分步骤画图，不然可能会有重叠的效果，影响分析。
'''

```

