Python大作业：疫情数据分析

2019213678 谭海川 2019211301班

目录：

一、作业概述

找一个有全球新冠病毒数据的网站，爬取其中的数据（禁止使用数据接口直接获取数据）。

要求爬取从 2021 年 12 月 5 日开始的连续 15 天的数据，国家数不少于 100 个。

1. 标明你的数据来源：包括网址和首页截图

2. 数据分析和展示应包括：

1) 15 天中，全球新冠疫情的总体变化趋势；

2) 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；

3) 累计确诊数排名前 10 的国家名称及其数量；

4) 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）；

5) 累计确诊人数占国家总人口比例最高的 10 个国家；

6) 疫苗接种情况（至少接种了一针及以上），请用地图形式展示；

7) 疫苗接种率（累计疫苗接种人数/国家人数）最低的 10 个国家；

8) 全球 GDP 前十名国家的累计确诊人数箱型图，要有平均值；

9) 死亡率最高的 10 个国家；

10) 其它你希望分析和展示的数据。

以上图形应包括完整的坐标、刻度、标签、图例等，如有必要请配上说明文字，对 图中的内容进行解释。

3. 根据以上数据，列出全世界应对新冠疫情最好的 10 个国家，并说明你的理由。

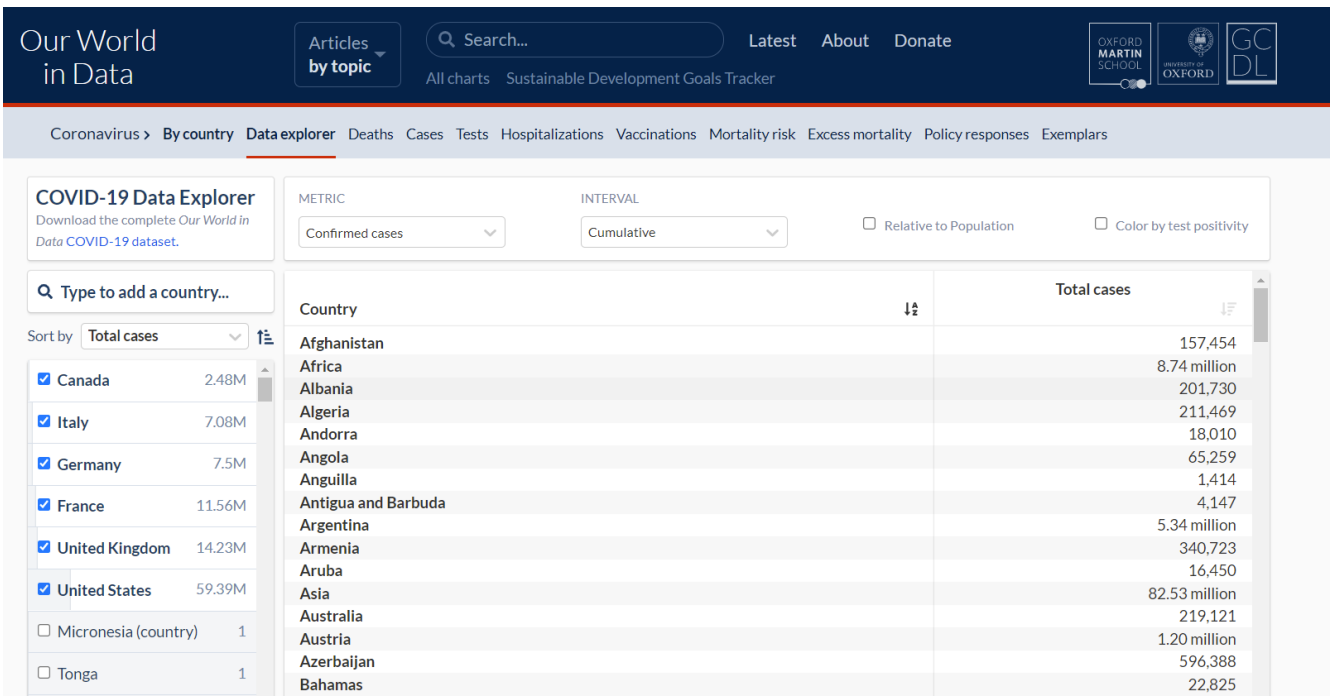
4. 针对全球累计确诊数，利用前 10 天采集到的数据做后 5 天的预测，并与实际数据进行 对比。说明你预测的方法，并分析与实际数据的差距和原因。

二、数据爬取

1.数据来源

来自网站：点击打开：

> **COVID-19 Data Explorer**
>
> Research and data to make progress against the world’ s largest problems
>
> https://ourworldindata.org/explorers/coronavirus-data-explorer?tab=table&zoomToSelection=true&time=2021-12-05&f…



该网站有按日计算的各个国家的：

- 确诊人数
- 新增确诊人数
- 死亡病例数
- 接种疫苗人数

数据齐全，可以通过之前学习的爬虫技术进行爬取

2.爬虫实现

分析网址

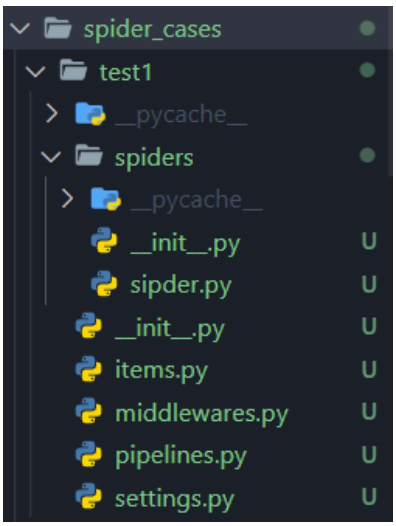
首先分析网页和网址结构，想办法找到历史数据。

观察某天的网址发现：

> https://ourworldindata.org/explorers/coronavirus-data-explorer?tab=table&zoomToSelection=true&time=2021-12-05&facet=none&uniformYAxis=0&pickerSort=asc&pickerMetric=total_cases&Metric=Confirmed+cases&Interval=Cumulative&Relative+to+Population=false&Color+by+test+positivity=false

- Time字段可以输入日期，而metric字段可以选择网页展示的是确诊病例数/死亡数/疫苗接种数
- 由此，可以对于2021-12-05到2021-12-19的连续15天分别构造相应的url，按照这些url进行爬取。

爬虫结构



spider.py是核心代码，其余的部分和之前提交的爬虫代码大同小异，篇幅限制不再附代码。

爬虫核心代码spider.py

- 由于该网站还统计了诸如 世界 高收入国家 低收入国家 亚洲 欧洲等非国家级别的数据，所以我在爬取的时候进行判断，去掉这些数据。
- 对于这15天，会自动填充相应的url进行爬取

- **由于网站中数据超过1million和1billion的数据是用的million和billion表示，所以在爬取的时候分别乘1e6和1e9进行转化。但是这样导致的结果之一是很多国家统计的人数精度不够，因此，后续数据分析过程多数时候单位采用的是 百万人**

Python

```python
import scrapy
from scrapy.http import response
from test1.items import Test1Item
not_country=["World","High income","Upper middle income","Asia","Europe","Lower middle income","North America","European Union","South America","Africa","Oceania"]
class mySpider(scrapy.spiders.Spider):
    name="bupt"
    allowed_domains=["ourworldindata.org"]
    with open("./date.txt","r") as f:
        date=f.read()
        #对于不同的15天，会自动修改对应的网址进行爬取
    start_urls=["https://ourworldindata.org/explorers/coronavirus-data-explorer?tab=table&zoomToSelection=true&time=2021-12-{}&facet=none&uniformYAxis=0&pickerSort=asc&pickerMetric=total_cases&Metric=Confirmed+cases&Interval=Cumulative&Relative+to+Population=false&Color+by+test+positivity=false".format(date)]
    def start_requests(self):
        for url in self.start_urls:
            yield scrapy.Request(url,callback=self.parse,cb_kwargs={"page_num":1})
    def parse(self,responce,**kwargs):
        item=Test1Item()
        for each in responce.xpath("/html/body/main/div/div[3]/div/div[1]/div/table/tbody/*"):
            item['country']=each.xpath("./td[1]/text()").get()
            cases=str(each.xpath("./td[2]/text()").get())
            cases=cases.replace(",","")
            #以下是对于输入为million的处理
            if (cases.find("million")!=-1):
                cases=cases.replace("million","")
                item['total_cases']=int((float(cases)*1000000))
            else :
                item['total_cases']=int(cases)
            if (item['country']and item['country'] not in not_country):
                yield(item)
```

三、数据分析

1.用到的库

`csv` 库是用来进行csv文件的读入和输出

`scipy` 库是用来进行科学计算的库，我用来计算线性回归进行预测。

`pyecharts` 库是用来进行画图的库

Python

```python
import pyecharts.options as opts
from pyecharts.charts import Line
from pyecharts.charts import Pie
from pyecharts.charts import Map
from pyecharts.charts import Boxplot
import scipy.stats as st
import csv
```

2.图表工具pyecharts

pyecharts 是一个用于生成 Echarts 图表的类库。Echarts 是百度开源的一个数据可视化 JS 库。用 Echarts 生成的图可视化效果非常棒，pyecharts 与 Python 进行对接，方便在 Python 中直接使用数据生成图。

3.数据读入

- cases是存储每个国家每天病例总数的字典，从爬取的csv文件中读取即可
- 死亡病例数，疫苗接种人数等读入的方法和总病例数是一样的。

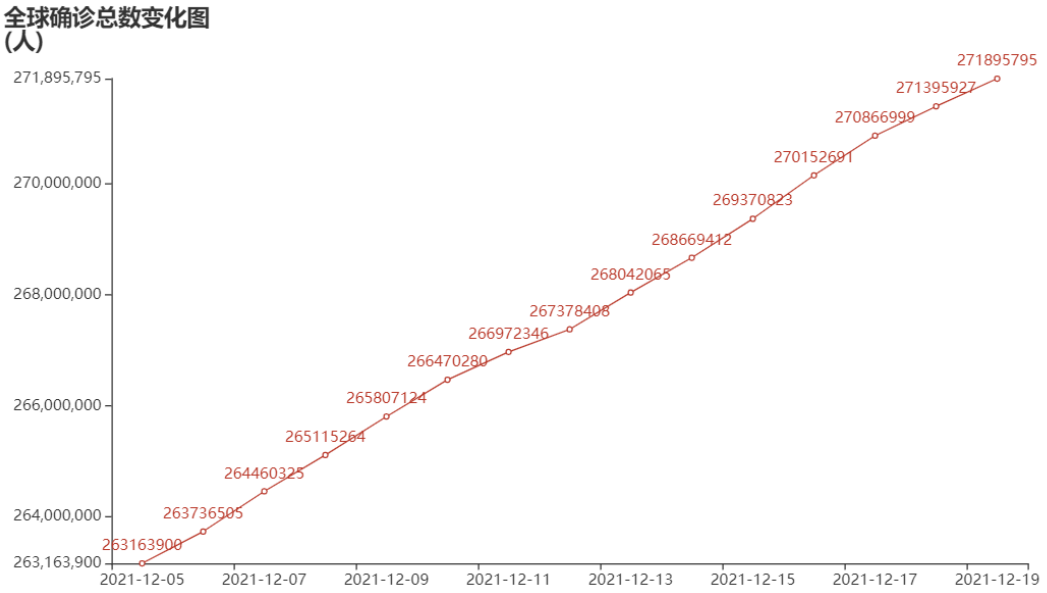
Haskell

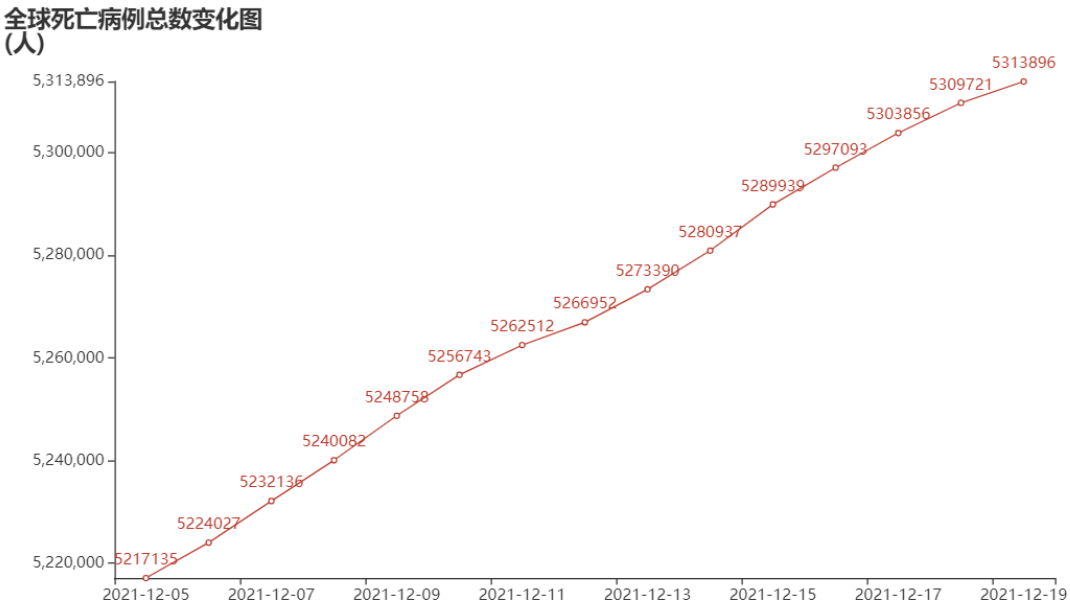
```
cases={}
country_list=[]
for date in date_list:
    with open("./spider_cases/"+date+".csv","r") as f:
        data=f.readlines()[1:]
        for line in data:
            if (len(line)!=0):
                t=line.split(",")
                cases[(t[0],date)]=int(t[1].strip())
            if (t[0] not in country_list):
                country_list.append(t[0])
death={}
for date in date_list:
    with open("./spider_death/"+date+".csv","r") as f:
        data=f.readlines()[1:]
        for line in data:
            if (len(line)!=0):
                t=line.split(",")
                death[(t[0],date)]=int(t[1].strip())
vaccinated={}
for date in date_list:
    with open("./spider_vaccinated/"+date+".csv","r") as f:
        data=f.readlines()[1:]
        for line in data:
            if (len(line)!=0):
                t=line.split(",")
                vaccinated[(t[0],date)]=int(t[1].strip())
new_cases={}
for date in date_list:
    with open("./spider_new/"+date+".csv","r") as f:
        data=f.readlines()[1:]
        for line in data:
            if (len(line)!=0):
                t=line.split(",")
                new_cases[(t[0],date)]=int(t[1].strip())
```

4.按要求进行数据分析

(1)

> 1) 15 天中，全球新冠疫情的总体变化趋势；





- 以上两张图分别从全球总确诊数量，总死亡病例数量出发，分别绘制对应的折线图。
- 可以看出，全球感染数量正在以较为稳定的速度快速增长，疫情仍在快速蔓延。
- 代码：

Python

```python
# 1
#统计确诊总数
total_cases=[]
for date in date_list:
    tot=0
    for country in country_list:
        tot+=cases[(country,date)]
    total_cases.append(tot)
#导出图表
chart = (
```

```
    Line()
    .add_xaxis(date_list)
    .add_yaxis("",total_cases)
    .set_global_opts(
        title_opts=opts.TitleOpts(title="全球确诊总数变化图"),
        yaxis_opts=opts.AxisOpts(
                type_="value",
                min_=min(total_cases),
                max_=max(total_cases)
            )
        )
    .render("全球确诊变化.html")
)
#统计死亡病例数
total_death=[]
for date in date_list:
    tot=0
    for country in country_list:
        tot+=death[(country,date)]
    total_death.append(tot)
#导出图表
chart = (
    Line()
    .add_xaxis(date_list)
    .add_yaxis("",total_death)
    .set_global_opts(
        title_opts=opts.TitleOpts(title="全球死亡病例总数变化图"),
        yaxis_opts=opts.AxisOpts(
                type_="value",
                min_=min(total_death),
                max_=max(total_death)
            )
        )
    .render("全球死亡病例数变化.html")
)
```

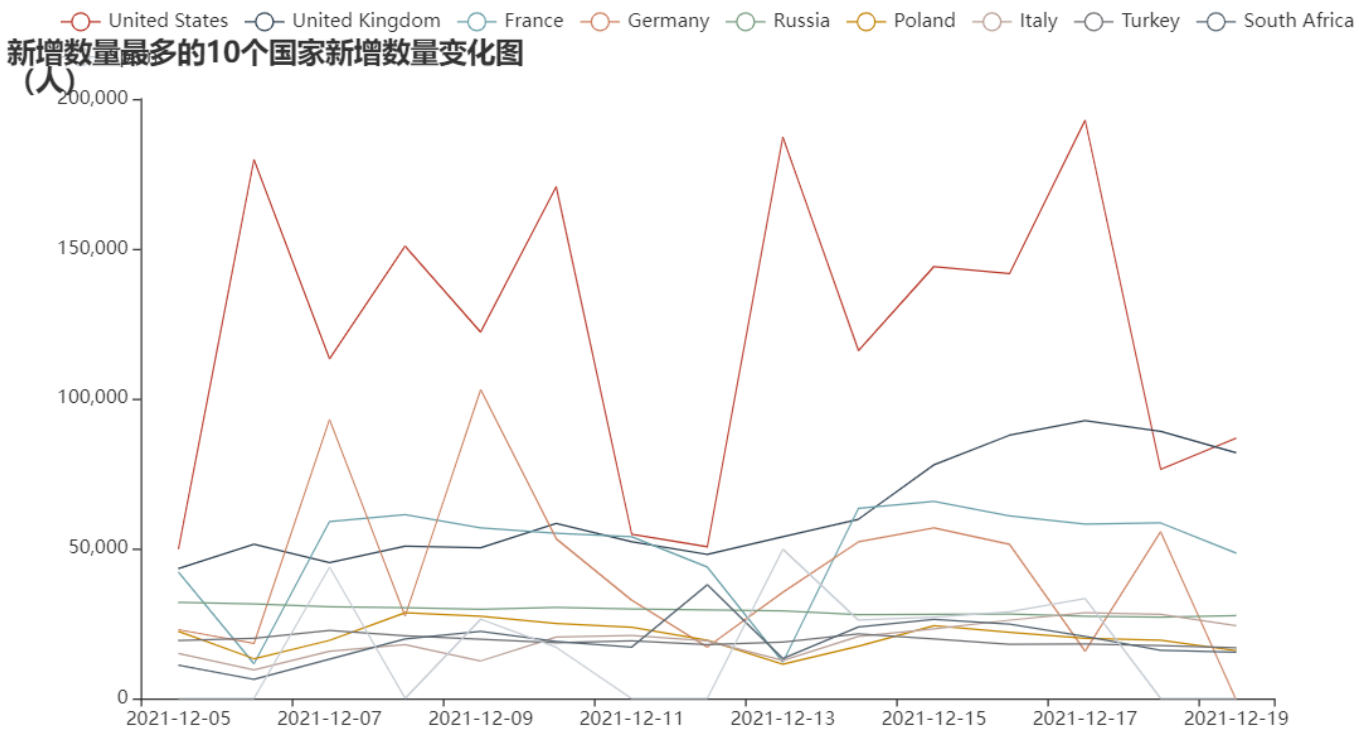
(2)

> 2) 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；

首先需要找出这个10个每天新增确诊数量最多的国家，只需要用19日的数据减去5日的数据就好。

之后根据每天的新增数量绘制折线图。

- 这新增最多的十个国家如图例所示：



- 可以看出，美国单日新增高居榜首，英法德等东欧国家紧随其后。
- 其次还有南非，俄罗斯，土耳其等区域大国。

代码：

Python

```python
# 2
increase_in_15_days=[]
for country in country_list:
    tot=0
    for date in date_list:
        tot+=new_cases[(country,date)]
    increase_in_15_days.append((tot,country))
#对于15天的新增数量进行排序
increase_in_15_days.sort(reverse=True)
increase_top_10_list=[]
#找到新增最多的十个国家
for i in range(10):
    increase_top_10_list.append(increase_in_15_days[i][1])
#导出图表
chart = (
    Line()
    .add_xaxis(date_list)
    .set_global_opts(
        title_opts=opts.TitleOpts(title="\n新增数量最多的10个国家新增数量变化图\n（人）"),
        yaxis_opts=opts.AxisOpts(
                type_="value",
                min_=0,
                max_=200000
            )
        )
)
#添加每个国家的折线图
for country in increase_top_10_list:
    increase=[]
    for date in date_list:
        t=new_cases[(country,date)]
        increase.append(t)
    chart.add_yaxis(country,increase,is_symbol_show=False)
chart.render("新增top10.html")
```

(3)

> 3）累计确诊数排名前 10 的国家名称及其数量；

- 这个分析比较简单，只需要统计各国确诊数量并排序输出即可。
- 结果如下：

Country	Total Cases
United States	50.89 million
India	34.75 million
Brazil	22.22 million
United Kingdom	11.38 million
Russia	10.04 million
Turkey	9.17 million
France	8.67 million
Germany	6.81 million
Iran	6.17 million
Spain	5.46 million

- 可以发现累计确诊前十名的国家和新增前十名的国家基本差不多
- 还是美国和各个区域的大国。

代码：

Python

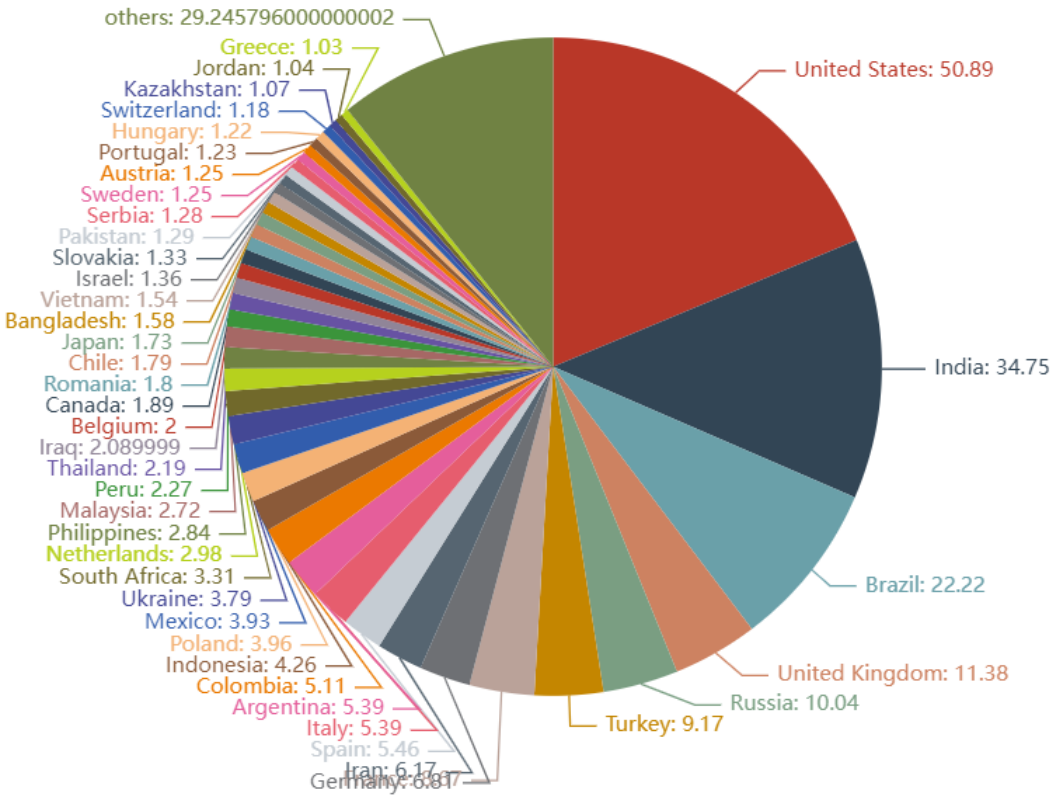
```python
#3
total_cases_dict=[]
for country in country_list:
    tot=cases[(country,date_list[-1])]
    total_cases_dict.append((tot,country))
#对确诊数量进行排序
total_cases_dict.sort(reverse=True)
#导出csv
cases_top_10_list=[]
with open("./累计确诊top10.csv","w",newline="") as f:
    writer=csv.writer(f)
    writer.writerow(["Country","Total Cases"])
    for i in range(10):
        t=total_cases_dict[i][0]
        t/=1000000
        writer.writerow([total_cases_dict[i][1],str(t)+" million"])
```

(4)

> 4) 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）；

- 这里对于确诊人数少于一百万的国家合并处理了
- 可以看到美国，印度和巴西总确诊人数占到了全球的三分之一以上，疫情十分严重。

全球累计确诊饼状图（单位 百万人）



代码：

Python

```python
# 4
#统计各国的疫情确诊比例
total_cases_dict_rev=[]
others=0
for i in total_cases_dict:
    if (i[0]>1000000):
        total_cases_dict_rev.append((i[1],i[0]/1000000))
    else:
        others+=i[0]/1000000 #小于一百万的合并处理
total_cases_dict_rev.append(("others",others))
#导出图表
chart = (
    Pie()
    .add("", total_cases_dict_rev)
    .set_global_opts(
        title_opts=opts.TitleOpts(title="全球累计确诊饼状图（单位 百万人）"),
        legend_opts=opts.LegendOpts(type_="scroll", pos_left="80%", orient="vertical",is_show=False)
    )
    .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
    .render("全球累计确诊饼状图.html")
)
```

(5)

> 5) 累计确诊人数占国家总人口比例最高的 10 个国家；

- 计算方法是用确诊人数除以国家人口
- 可以发现有的国家确诊比例甚至已经达到四分之一。

Country	Cases Ratio
Andorra	26.5645401072975 25%
Montenegro	25.633983119259046%
Slovakia	24.3557581498853 36%
Seychelles	24.312492417195777%
Georgia	22.84456494290492%
Slovenia	21.57871848306942%
San Marino	20.5985242672781 26%
Lithuania	18.675493389623707%
Estonia	17.5160449295758 73%
Netherlands	17.352721253164617%

Python

```python
#5
case_ratio=[]
case_ratio_country={}
for country in country_list:
    #统计确诊比例
    tot=cases[(country,date_list[-1])]/population[country]*100
    case_ratio.append((tot,country))
    case_ratio_country[country]=tot
#对于确诊比例进行排序
case_ratio.sort(reverse=True)
#导出csv
with open("./确诊比例top10.csv","w",newline="") as f:
    writer=csv.writer(f)
    writer.writerow(["Country","Cases Ratio"])
    for i in range(10):
        t=case_ratio[i][0]
        writer.writerow([case_ratio[i][1],str(t)+"%"])
```

(6)

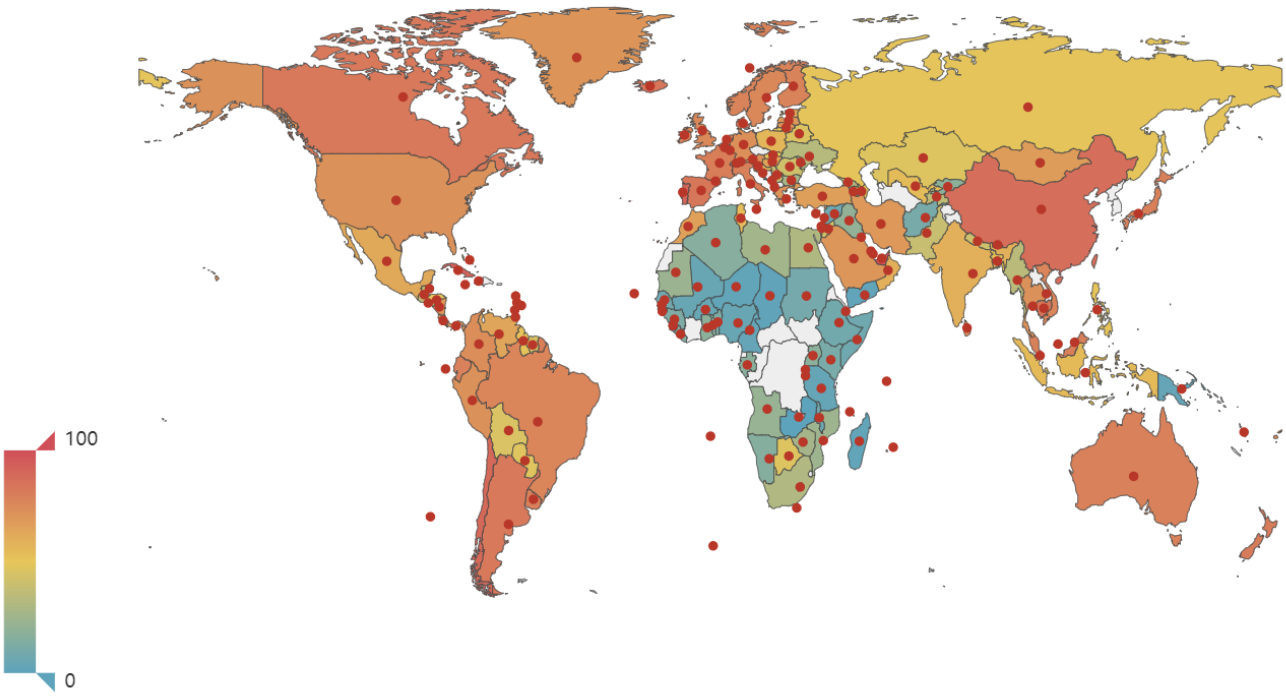
> 6) 疫苗接种情况（至少接种了一针及以上），请用地图形式展示；

这里使用了pyecharts的地图绘制功能

首先统计各个国家的疫情接种率，按照疫情接种率对各个国家进行染色。

红色表示接种率高，蓝色表示接种率低。

全球疫苗接种率地图



- 可以看出中国接种率很高，美国和欧洲等国家也很好
- 但是非洲国家接种率普遍较低。

Python

```python
# 6
vacc_ratio=[]
vacc_map=[]
vacc_ratio_country={}
#统计每个国家的疫苗接种率
for country in country_list:
    tot=vaccinated[(country,date_list[-1])]/population[country]*100
    vacc_map.append((country,tot))
    vacc_ratio.append((tot,country))
    vacc_ratio_country[country]=tot
#导出图表
chart = (
    Map()
    .add("", vacc_map, "world")
    .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="全球疫苗接种率地图"),
        visualmap_opts=opts.VisualMapOpts(max_=100),
    )
    .render("全球疫苗接种率地图.html")
)
```

(7)

> 7) 疫苗接种率（累计疫苗接种人数/国家人数）最低的 10 个国家；

- 用统计到的疫苗接种率数据，直接统计最低的10个国家输出即可

Country	Vaccinated Ratio
Burundi	0.03272018214289124%
Haiti	1.06206329491751%
Chad	1.6903473458593075%
Yemen	1.8256487892677884%
South Sudan	1.9967880866446928%
Niger	2.0183307212017816%
Madagascar	2.07196047409028 37%
Cameroon	2.96124064322765%
Papua New Guin	3.1134410423938563%
Tanzania	3.3821997785081 92%

- 经过验证，这些国家的经济水平较为落后。

Python

```python
# 7
#对接种率进行排序
vacc_ratio.sort()
#导出csv
with open("./接种率last10.csv","w",newline="") as f:
    writer=csv.writer(f)
    writer.writerow(["Country","Vaccinated Ratio"])
    cnt=0
    i=0
    while (cnt<10):
        i+=1
        t=vacc_ratio[i][0]
        if (t<1e-5):
            continue
        cnt+=1
        writer.writerow([vacc_ratio[i][1],str(t)+"%"])
```

(8)

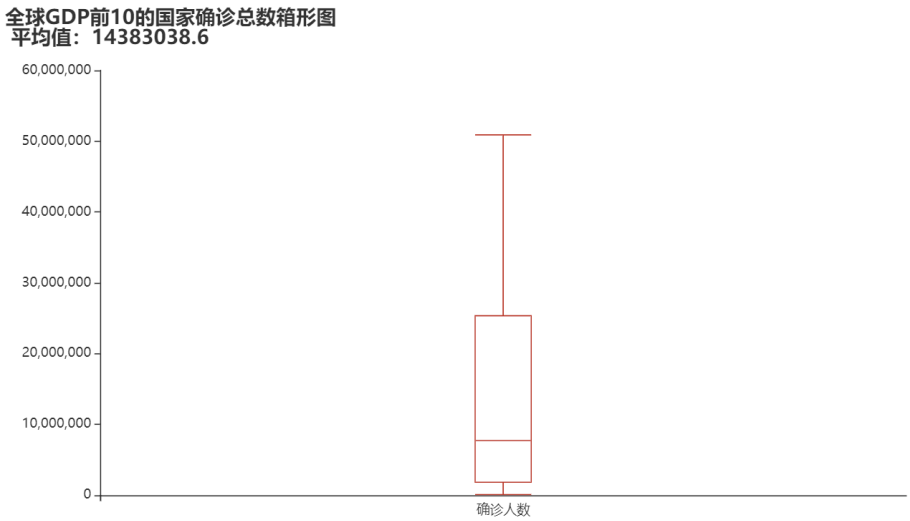
> 8) 全球 GDP 前十名国家的累计确诊人数箱型图，要有平均值；

这需要先查出GDP前十名的国家有哪些：

Python

```
gdp_top_10_country=["United States","China","Japan","Germany","India","United Kingdom","France","Brazil","Italy","Canada"]
```

之后统计这些国家的确诊人数，用pyecharts绘制箱型图，同时计算平均值并输出。



- 可以看出整体数据跨度非常大，最低的是中国，最高的是美国。
- 平均值是一千四百万人，不过中位数缺不到一千万，平均值是由美国拉上去的。
- 代码：

Python

```
# 8
gdp_top_10_country=["United States","China","Japan","Germany","India","United Kingdom","France","Brazil","Italy","Canada"]
box_data=[]
#统计这些国家的确诊总数
for country in gdp_top_10_country:
    box_data.append(cases[(country,date_list[-1])])
#导出图表
c = Boxplot()
c.add_xaxis(["确诊人数"])
c.add_yaxis("", c.prepare_data([box_data]))
c.set_global_opts(title_opts=opts.TitleOpts(title="全球GDP前10的国家确诊总数箱形图\n 平均值：{}".format(sum(box_data)/len(box_data))))
c.render("GDP前10的国家确诊总数箱形图.html")
```

(9)

> 9) 死亡率最高的 10 个国家；

用死亡病例数字除以总人数，就得到死亡率，由此统计即可。

Country	Death Ratio
Peru	26.564540107297525%
Bulgaria	25.633983119259046%
Bosnia and Herze	24.355758149885336%
Hungary	24.3124924171957777%
Montenegro	22.844564942904 92%
North Macedonia	21.57871848306942%
Georgia	20.5985242672781 26%
Romania	18.675493389623707%
Croatia	17.5160449295758 73%
Slovakia	17.352721253164617%

代码：

Python

```python
# 9
death_ratio=[]
death_ratio_country={}
for country in country_list:
    tot=death[(country,date_list[-1])]/population[country]*100
    death_ratio.append((tot,country))
    death_ratio_country[country]=tot
#按照死亡率进行排序
death_ratio.sort(reverse=True)
#导出csv
with open("./死亡比例top10.csv","w",newline="") as f:
    writer=csv.writer(f)
    writer.writerow(["Country","Death Ratio"])
    for i in range(10):
        t=case_ratio[i][0]
        writer.writerow([death_ratio[i][1],str(t)+"%"])
```

可以看出死亡率较高的国家仍然是一些疫苗接种率低，并且经济发展水平低的国家。

（10）

> 根据以上数据，列出全世界应对新冠疫情最好的 10 个国家，并说明你的理由。

考虑已有数据，可以通过综合死亡率，确诊率，疫苗接种情况来综合设计一个评分，来评估一个国家的防疫现状。

我认为一个国家的死亡率越低，确诊率越低，疫苗接种率越高，可以说明这个国家应对疫情最好，所以对死亡率，确诊率和疫苗接种率加权处理得到评分。

我设计的一个评分标准是： $score = 疫苗接种率 - 5 * 确诊比例 - 5 * 死亡率 + 50$

这个标准不一定很科学，仅作为尝试参考。

这样得到一个分数，取分数最高的十个国家：

Country	Score
China	274.42
Cambodia	260.66
New Zealand	260.15
South Korea	258.21
Taiwan	255.2
Brunei	248.39
Bhutan	247.83
Australia	247.71
Japan	245.11
Vietnam	238.24

代码：

Python

```python
score=[]
score_map=[]
#对每个国家计算分数
for country in country_list:
    s=50+vacc_ratio_country[country]-5*case_ratio_country[country]-5*death_ratio_country[country]
    score.append((s,country))
    score_map.append((country,275-s))
score.sort(reverse=True)
#导出csv
with open("./抗疫最好的top10.csv","w",newline="") as f:
    writer=csv.writer(f)
    writer.writerow(["Country","Score"])
    for i in range(10):
        t=score[i][0]
        writer.writerow([score[i][1],str(t)])
```

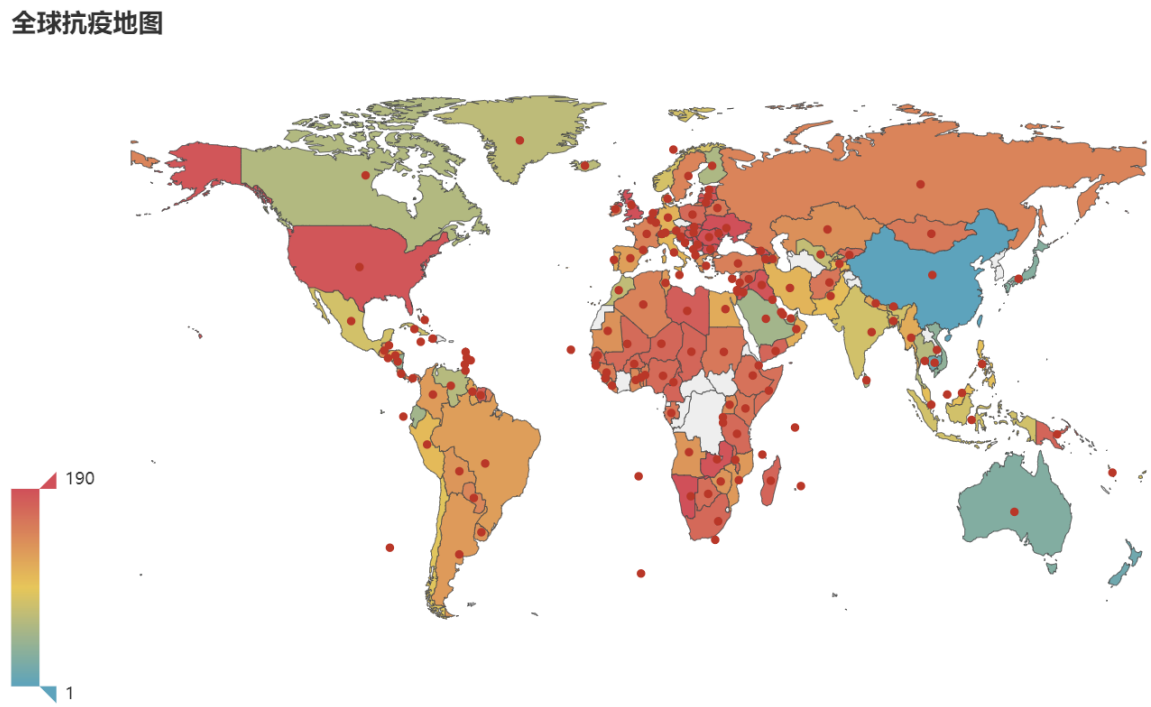
(11)

> 10) 其它你希望分析和展示的数据。

我选择了两个角度：

- **根据上一题设计的评分标准，分析并绘制全球国家抗疫水平地图：**
- **绘制确诊比例和疫苗接种率的散点图，分析其关联**

全球国家抗疫水平地图



从这张图中可以显著看出，中国防疫取得了非常大的成果

而美国和欧洲防疫状况堪忧。

代码：

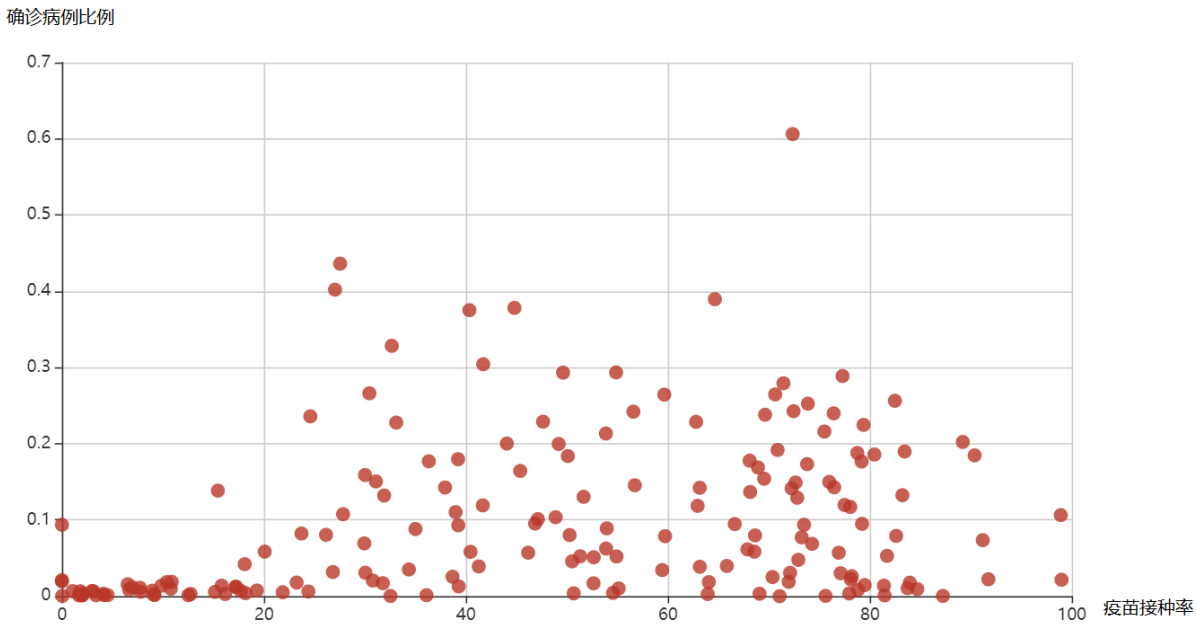
Python

```python
for i in score_map:
    maxx=max(maxx,i[1])
    minn=min(minn,i[1])
#导出图表
chart = (
    Map()
    .add("", score_map, "world")
    .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="全球抗疫地图"),
        visualmap_opts=opts.VisualMapOpts(max_=190,min_=minn),
    )
    .render("全球抗疫地图.html")
)
```

确诊比例和疫苗接种率散点图

为了研究确诊比例和疫苗接种率之间的关系，我将这所有的国家的确诊比例和疫苗接种率绘制成散点图，纵轴是确诊比例。横轴是疫苗接种率。

得到结果如下:



可以发现，疫苗接种率和确诊病例的比例并没有十分明确的关系。

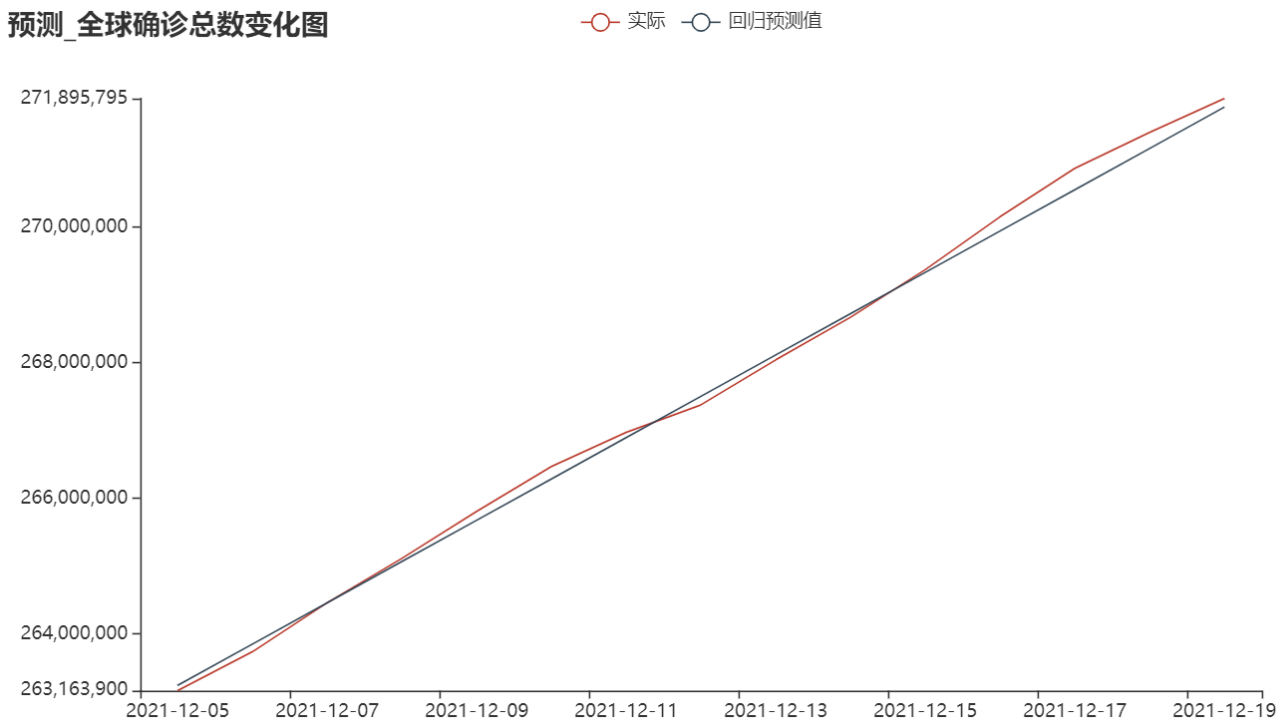
分析：

- 疫苗接种可以部分算作是疫情蔓延的结果，可能很多国家是因为疫情严重，其他防控措施无效，大力推广疫苗，所以并没有预期的疫苗接种率越高，患病率越低的关系。
- 相比起疫苗，政府的防控措施，民众意识，地理位置，医疗水平，气候等因素对于疫情发展的影响也非常大，所以疫苗接种的影响不容易直观的看出来
- 新冠病毒有很多如奥密克戎的变种，已经接种的疫苗对于各种变种的免疫力有所下降甚至消失，这也是疫苗作用不明显的原因。

(12)

> 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。说明你预测的方法，并分析与实际数据的差距和原因。

- 观察全球确诊病例总数变化图，发现增长很稳定，接近线性函数
- 于是尝试使用线性回归进行预测
- 使用了scipy库的线性回归函数
- 结果如下：



其中，红线表示的是实际的确诊人数，而灰色的线表示的是回归的曲线。

这条灰色的曲线是由前十天的数据计算得到的，可以看出对于后五天的数据预测结果还不错。

但是仍需意识到，疫情的传播速度和自然状况，基因突变，各国政策息息相关，所以很难用一个简单的方法来预测疫情的发展。

四、实验总结

经过这次实验，我学会了使用python进行简单的数据分析，对于python常用库和函数的使用也更加熟练，对于Python的解释器也有了更加深入的理解。这次实验我从多个方面入手，通过爬虫爬取的数据，加以整理。最终以图表的形式呈现出来。我深深感受到了数据分析和统计的重要性，也感受到了python这个工具对于数据分析所带来的便利性和便捷性。

这次统计疫情数据，让我意识到全球抗疫之路前路漫漫，让我意识到相对于疫情肆虐的西方国家，中国的抗疫举措多么有效，抗疫成果多么显著。我深深的为自己是一个中国人而感到自豪。

在今后的学习生活中，我将注重自己对于信息获取数据分析方面技能的培养，将这种数据分析的思维和意识融入到生活中。同时进一步增强对于各种工具的使用的学习，学会从各个角度全面的思考和分析问题。