

队伍编号	dsa2300059
题号	B

人口数量的分析与预测：基于可视化展示与计算机模拟的研究

摘 要

本题主要研究的问题是：对于人口数量的分析与预测。通过赛方给定相关数据以及团队自身收集的数据，分析统计所有国家的人口增长率与增长人数，并找出相对应较为突出的国家。比较不同国家的人口数量。对指定国家的人口年龄构成进行分析以及预测世界、中国与印度的未来人口，而本论文主要运用 **Excel**、**R** 语言、**Python** 与 **MATLAB** 进行分析与画图。

对于问题一，事先运用 **Excel** 的筛选功能，删去较小的岛屿以及地区构成人数。通过 **Python** 的 **openpyxl** 库，读入相关人口数据并对其人口增长率与增长人数进行计算。随后运用两个字典分别储存各个国家的人口增长率以及增长人数，进行排序，输出结果。然后用 **Excel** 提取出世界人口的变化，最终使用 **pandas** 再次读入数据，运用 **matplotlib** 进行绘图。

对于问题二，用有关数据文件进行筛选，利用 **Python** 的 **pandas** 库筛选出年份为 2021 的数据，之后在 **R** 语言中进行分析绘制。利用 **ggplot2** 包绘制所有国家的纵向条形图。但国家之间人口数量差距过大，因此，独立将人口数排名前 20 名国家，以及人口数后 20 名国家单独作纵向条形图来着重表现。

对于问题三，基于已知数据，我们选择了人口数量具有相同量级的国家：巴西、俄罗斯、日本，并对其进行类比分析。在本题的解决过程中，我们主要使用 **Excel** 对原数据进行分组、聚合等处理，以得到三个国家不同年龄段分布的相关数据，再通过 **R** 语言进行可视化展示，并结合三个国家的国情分析各国不同年龄段人口的分布原因。

对于问题四，查阅相关文献，并结合具体情况，得知印度以及世界人口还有增长空间，采用 **Logistic** 人口生长模型进行曲线拟合，编程工具为 **MATLAB** 的曲线拟合工具箱，由此得以预测未来的人口增长。对于中国的人口预测，则转而采用计算机模拟的方法，通过各个年龄的生育率、人口、男性女性人口、死亡率等进行参数模拟，利用二项分布模拟各个年份各个年龄段生育的人员以及死亡人员。最后依据不同的生育率，得到对应的预测曲线，利用 **Python** 进行画图。

关键词：Excel R 语言 Python MATLAB Logistic 人口生长模型 计算机模拟 曲线拟合 二项分布

目录

1.问题的背景与重述	1
1.1 问题的背景	1
1.2 问题的重述	1
2.问题处理流程	1
3.问题一的解决方案	3
3.1 数据预处理	3
3.2 数据统计以及数据分析	3
4.问题二的解决方案	7
4.1 数据处理	7
4.2 数据可视化展示	7
4.2.1 人口数量前 20 国家与人口数量后 20 国家数据	7
4.2.2 人口数量前 20 国家与人口数量后 20 国家纵向柱形图	9
4.2.3 人口数量前十国家的地理图	10
5.问题三的解决方案	10
5.1 数据预处理	10
5.2.数据可视化展示与原因分析.....	10
5.2.1 人口分布直方图	10
5.2.2 人口堆叠面积图	11
5.2.3 人口年龄比率折线图	12
5.2.4 年龄分布原因解释	12
5.问题四的解决方案	13
5.1 理论准备	13
5.2 参数选择以及模型构建	14
5.2.1 世界人口部分	14
5.2.2 印度人口部分	15
5.2.3 中国人口部分	15
6.总结以及建议	17
7.参考文献	18
8.附录	19
附录 1: 统计各国人口增长率及其增长人口.py.....	19
附录 2: 统计平均人口增长率及其绝对数.py	19
附录 3: 绘制总人口曲线变化图.py	21
附录 4: 全球人口画图以及相关数据导出.py	21
附录 5: 印度人口画图以及相关数据导出.py	22
附件 6: 中国人口画图.py.....	22
附录 7: 中国人口预测（无措施） py	23
附录 8: 中国人口预测（有措施） py	25
附录 9: 问题三部分画图 R 代码	28
人口堆叠柱状图	28
折线图	28

直方图	29
附录 10: 各个国家人口数据排名	30

1.问题的背景与重述

1.1 问题的背景

随着人类的不断发展和进步，全球人口数量也在不断增长。据联合国人口基金会的数据显示，截至 2021 年，全球人口已经超过了 78 亿，而这个数字预计将在未来几十年内继续增长。人口增长对于全球社会、经济和环境都带来了巨大的影响，因此对人口数量和结构进行预测和分析是非常重要的。

从国家的角度来说，人口预测和分析可以帮助政府和决策者了解人口数量和结构的变化趋势，从而制定相应的政策和计划。并且，人口预测和分析也可以帮助企业 and 市场研究人员了解人口的消费习惯和需求变化，从而为企业决策和市场营销提供依据。而对于学术方面而言，人口预测和分析也可以帮助学者和研究人员了解人口结构对社会、经济和环境的影响，从而为未来的发展提供参考。例如，如果预测到城市化趋势将继续加速，研究人员可以探讨城市化对环境和资源的影响，并提出相应的解决方案。

因此，本届全国大学生数据统计与分析竞赛深刻领悟时代思想，与时俱进，提出相对应问题，鼓励学生将所学运用到实际问题中，运用所学知识对于人口进行统计与分析。

1.2 问题的重述

问题 1: 运用相关数据，绘制 1950-2021 总人口变化趋势折线图并对其进行分析。分析统计人口增长率以及增长人数，分别进行排名得到各自的前十位以及最后十位。

问题 2: 就 2021 年的横截面数据，将不同国家的总人口数进行比较，绘制柱状图，并得出前十位以及最后十位的国家。

问题 3: 任选三个国家，绘制其人口的不同年龄段构成直方图，对其异同点进行对比分析，并分析其原因。

问题 4: 进行数学建模，对中国、印度以及世界的人口进行预测，最终预测到 2100 年，并对其变化趋势进行分析。

2.问题处理流程

在本章节中，主要采取思维导图方法进行分析
对于问题 1 与问题 2:

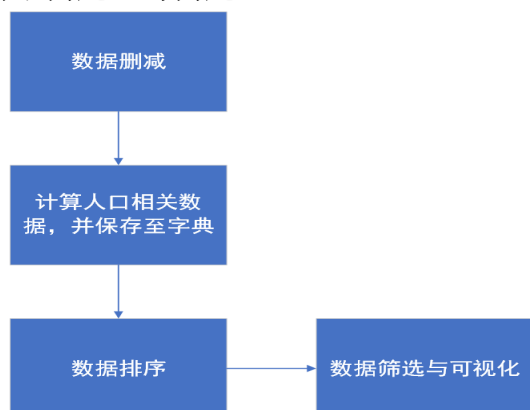


图 1 问题一处理导图

对于问题 2:

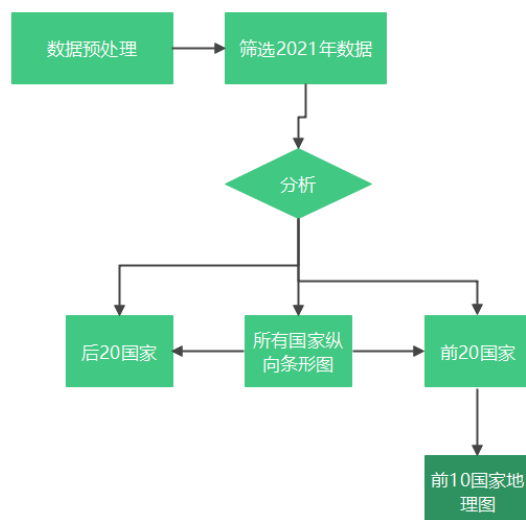


图 2 问题二处理导图

对于问题 3:

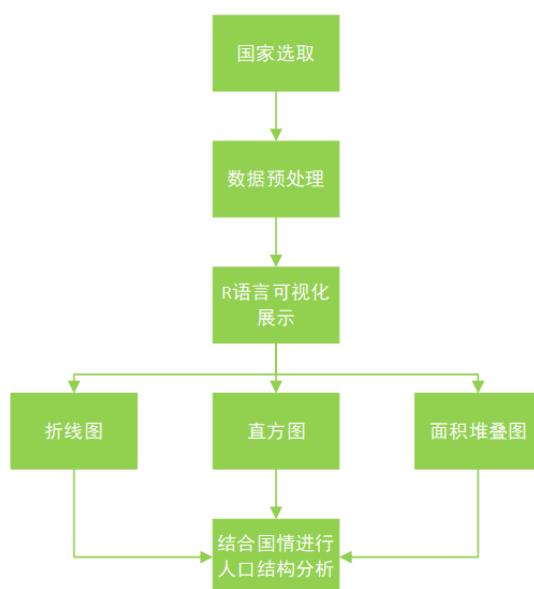


图 3 问题三处理导图

对于问题 4:

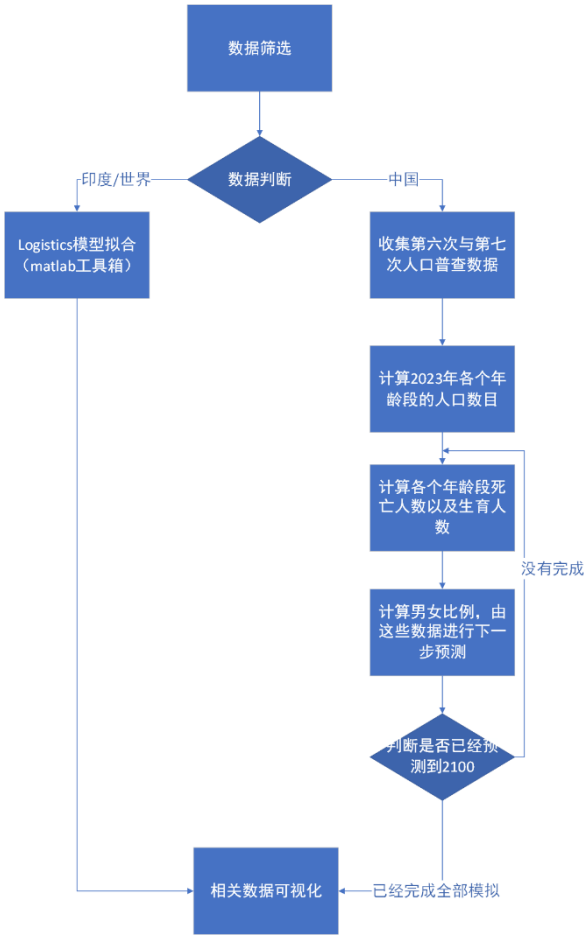


图 4 问题四处理导图

3.问题一的解决方案

3.1 数据预处理

首先，运用 Excel 的筛选功能，将“population.csv”几大洲、不同收入国家、小岛屿以及不同人口分类的国家群体等组成的人口数据删去（但是保留国家的海外属地），得到“各个国家人口数据.xlsx”，例子有：Africa (UN)、Asia (UN)以及 Europe (UN)等。

其次，运用 Python 程序（附录 1）对于表中的数据进行处理，得到不同国家的人口增长率以及增长人口（1950 年赋值 0 处理，没有以前的数据暂不考虑），输出得到“各个国家人口数据（已处理）.xlsx”文件。

最后，还是运用 Excel，将“population.csv”中的“World”部分筛选出来，保存其为“世界总人口变化.xls”，运用附录 3 程序进行图像绘制。

3.2 数据统计以及数据分析

运用相关文件以及附录 2 程序，循环读入，统计不同国家 2010-2021 年间的人口增长率以及增长人口，并最终除以 11，做平均值处理，得到平均人口增长率以及平均人口增长，经过排序输出后得到相对应的表格，用于后期可视化（R 语言程序包放于对应文

件夹中），并得到如下结果：

表 1 人口平均增长率前十位			
排名	英文名	中文名	增长率
1	Jordan	约旦哈希姆王国	0.0418
2	Qatar	卡塔尔	0.0395
3	Oman	阿曼	0.0394
4	Niger	尼日尔	0.0372
5	Equatorial Guinea	赤道几内亚共和国	0.0358
6	Mayotte	马约特（法属）	0.0357
7	Angola	安哥拉	0.0348
8	Democratic Republic of Congo	刚果民主共和国	0.0329
9	Chad	乍得共和国	0.0329
10	Maldives	马尔代夫共和国	0.0327

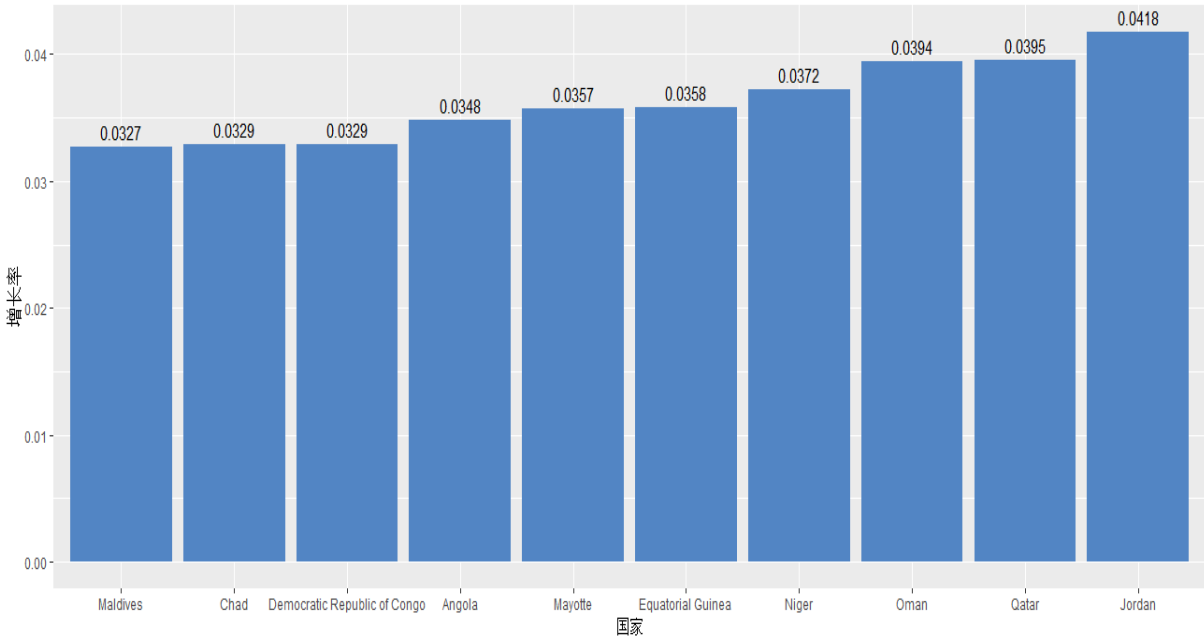


图 5 人口增长率前十图

表 2 人口平均增长绝对数前十位

排名	英文名	中文名	增长绝对数
1	India	印度	15177300
2	China	中国	7063827
3	Nigeria	尼日利亚联邦共和国	4768042
4	Pakistan	巴基斯坦	3358873
5	Ethiopia	埃塞俄比亚	2822292
6	Indonesia	印度尼西亚	2703363
7	Democratic Republic of Congo	刚果民主共和国	2682078
8	United States	美国	2346798
9	Egypt	埃及	2000887
10	Bangladesh	孟加拉人民共和国	1905918

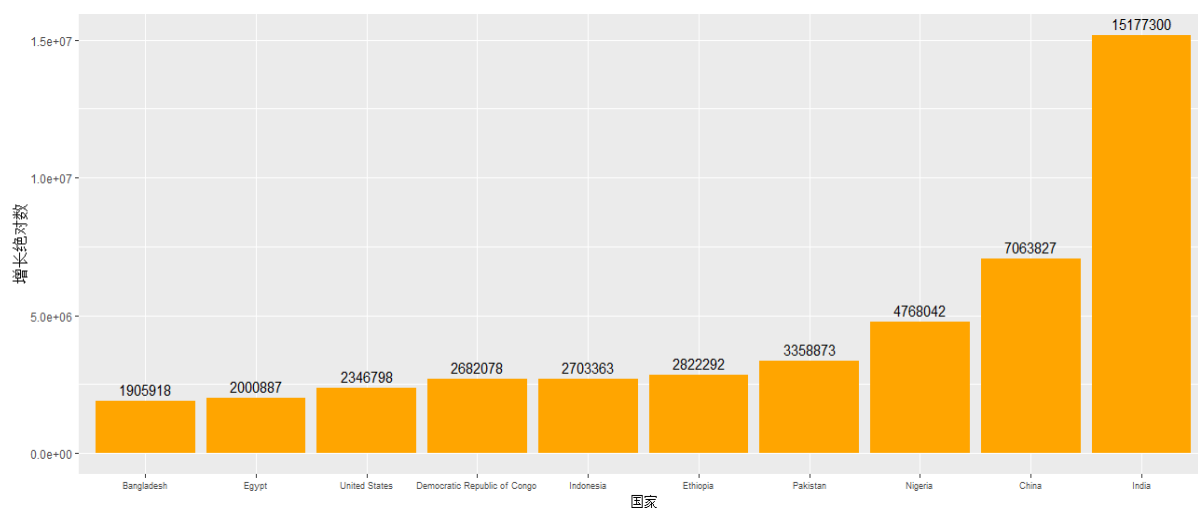


图 6 人口增长前十图

表 3 人口平均增长率最后十位表

排名	英文名	中文名	增长率
1	American Samoa	美洲萨摩亚	-0.0181
2	Moldova	摩尔多瓦	-0.0168
3	Bosnia and Herzegovina	波斯尼亚和黑塞哥维那	-0.014
4	Puerto Rico	波多黎各	-0.0121
5	Saint Martin (French part)	圣马丁（法属领地）	-0.0121
6	Wallis and Futuna	瓦利斯和富图纳	-0.0112
7	Lithuania	立陶宛	-0.0109
8	Montserrat	蒙特塞拉特（英属领地）	-0.0107
9	Latvia	拉脱维亚	-0.0105
10	Bulgaria	保加利亚	-0.0089

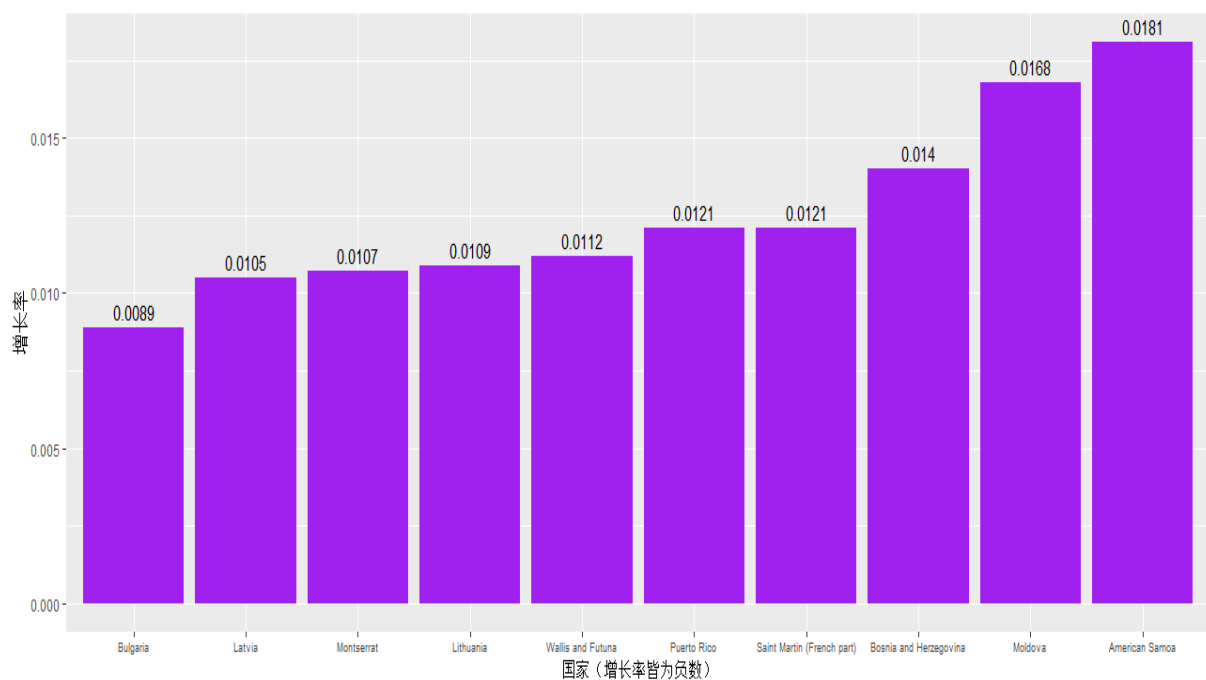


图 7 人口平均增长率最后十位图

表 4 人口平均增长绝对数最后十位

排名	英文名	中文名	增长绝对数
1	Japan	日本	-317536
2	Ukraine	乌克兰	-195600
3	Syria	叙利亚	-92108
4	Romania	罗马尼亚	-91514
5	Bulgaria	保加利亚	-64218
6	Moldova	摩尔多瓦	-56061
7	Greece	希腊	-53492
8	Italy	意大利	-52919
9	Bosnia and Herzegovina	波斯尼亚和黑塞哥维那	-49104
10	Venezuela	委内瑞拉	-46833

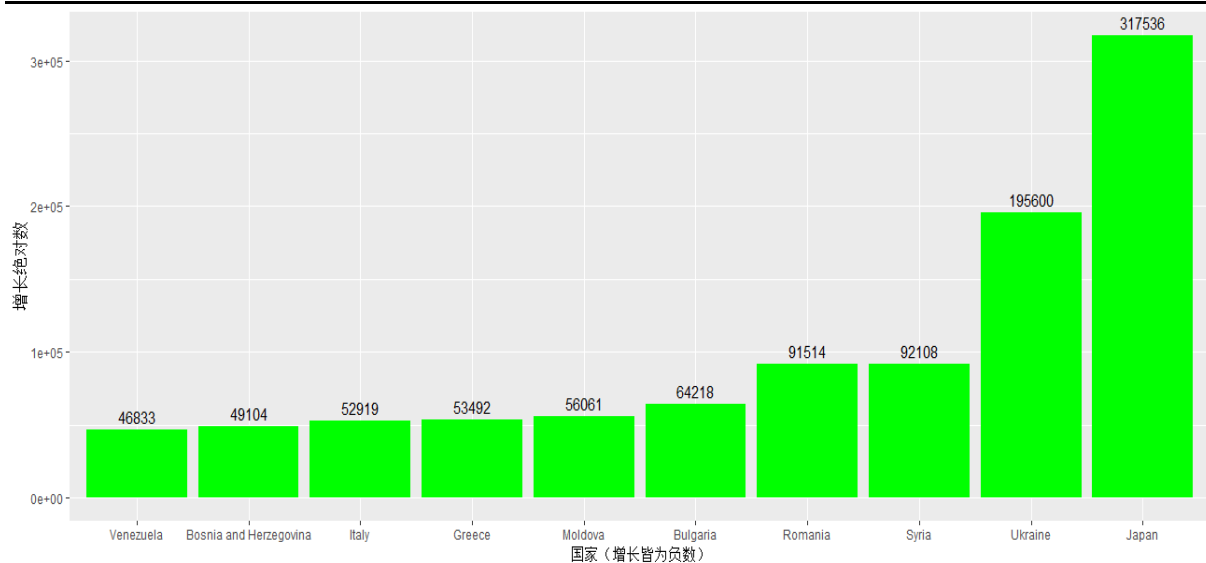


图 8 人口增长最后十位图

而经过绘图之后，世界人口的变化曲线如下

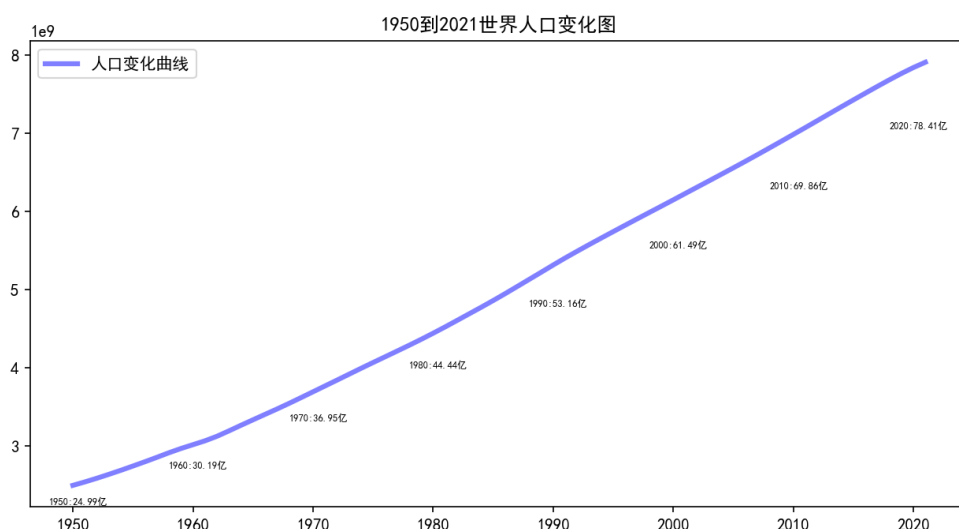


图 9 1950 到 2021 世界人口变化图

而由图我们可以看出，世界人口总体仍然处于攀升阶段，1950 年全球人口约为 25 亿，而到 2021 年则增加到了近 80 亿。这是一个非常显著的增长。同时根据前面表格我们可以看出来，世界人口的增长主要来源于亚非拉国家，而人口负增长主要来源于欧洲以及日本等地的发达国家，人口增长的地区不均衡依旧值得注意，它可能会在未来影响人口的进一步增长。

4.问题二的解决方案

4.1 数据处理

用“各个国家人口数据（已处理）.xlsx”文件进行筛选，利用 Python 的 pandas 库筛选出年份为 2021 的数据，另存为“2021 各国数据.csv”，之后再 R 语言中进行分析绘制。在 R 语言中，首先导入文件为 df1，进行按照人各国的人口数量进行升序排序，同时列出各个国家的排名，然后根据排名将国家分为三类，依次为前十名，后十名，以及其他国家。首先利用 ggplot2 包绘制所有国家的纵向条形图。但国家之间人口数量差距过大，因此，独立将前人口数前 20 名国家，以及人口数后 20 名国家单独作纵向条形图来着重表现。

4.2 数据可视化展示

4.2.1 人口数量前 20 国家与人口数量后 20 国家数据

由于图片过大，分割展示在附录 10 中。

由图并结合处理软件可以得出人口数量前 10 的国家，分别是中国，印度，美国，印度尼西亚，巴基斯坦，巴西，尼日利亚，孟加拉国，俄罗斯联邦，墨西哥。人口数量后 10 国家为安圭拉，瑙鲁，瓦利斯和富图纳，图瓦卢，加勒比海圣巴特岛，圣皮埃尔和密克隆，圣赫勒拿，蒙特塞拉特，纽埃，托克劳。

表 5 人口数量前 10 国家

排名	国家	中文名
1	China	中国
2	India	印度
3	United States	美国
4	Indonesia	印度尼西亚
5	Pakistan	巴基斯坦
6	Brazil	巴西
7	Nigeria	尼日利亚
8	Bangladesh	孟加拉国
9	Russia	俄罗斯联邦
10	Mexico	墨西哥

表 6 人口数量后 10 国家

排名	国家	中文名
213	Anguilla	安圭拉
214	Nauru	瑙鲁
215	Wallis and Futuna	瓦利斯和富图纳
216	Tuvalu	图瓦卢
217	Saint Barthelemy	加勒比海圣巴特岛
218	Saint Pierre and Miquelon	圣皮埃尔和密克隆
219	Saint Helena	圣赫勒拿
220	Montserrat	蒙特塞拉特
221	Niue	纽埃
222	Tokelau	托克劳

4.2.2 人口数量前 20 国家与人口数量后 20 国家纵向柱形图

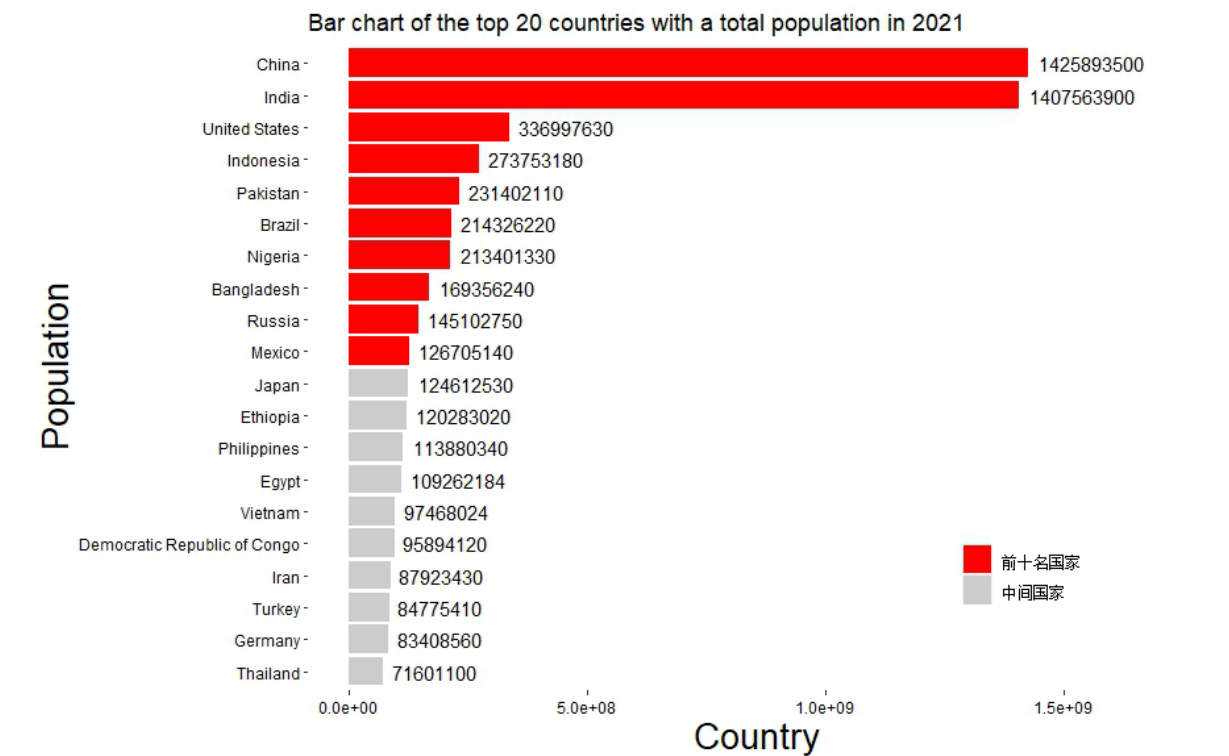


图 10 人口数量前 20 名国家

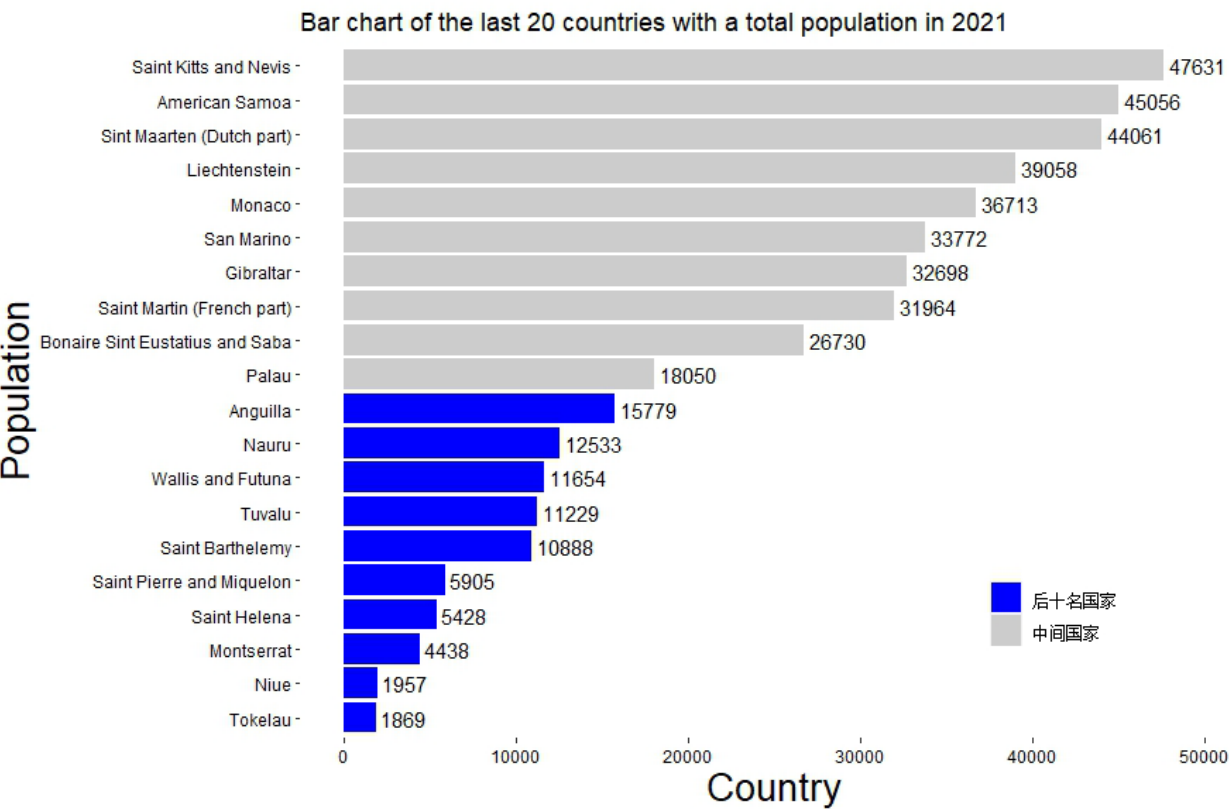


图 11 人口数量后 20 的国家

由图 1.2.1 与图 1.2.2 可知人口数前 10 名国家中，印度与中国相较于其他国家，人口数量差距十分巨大。而在，人口数量后 10 名国家中，人口数量差距不大。

4.2.3 人口数量前十国家的地理图

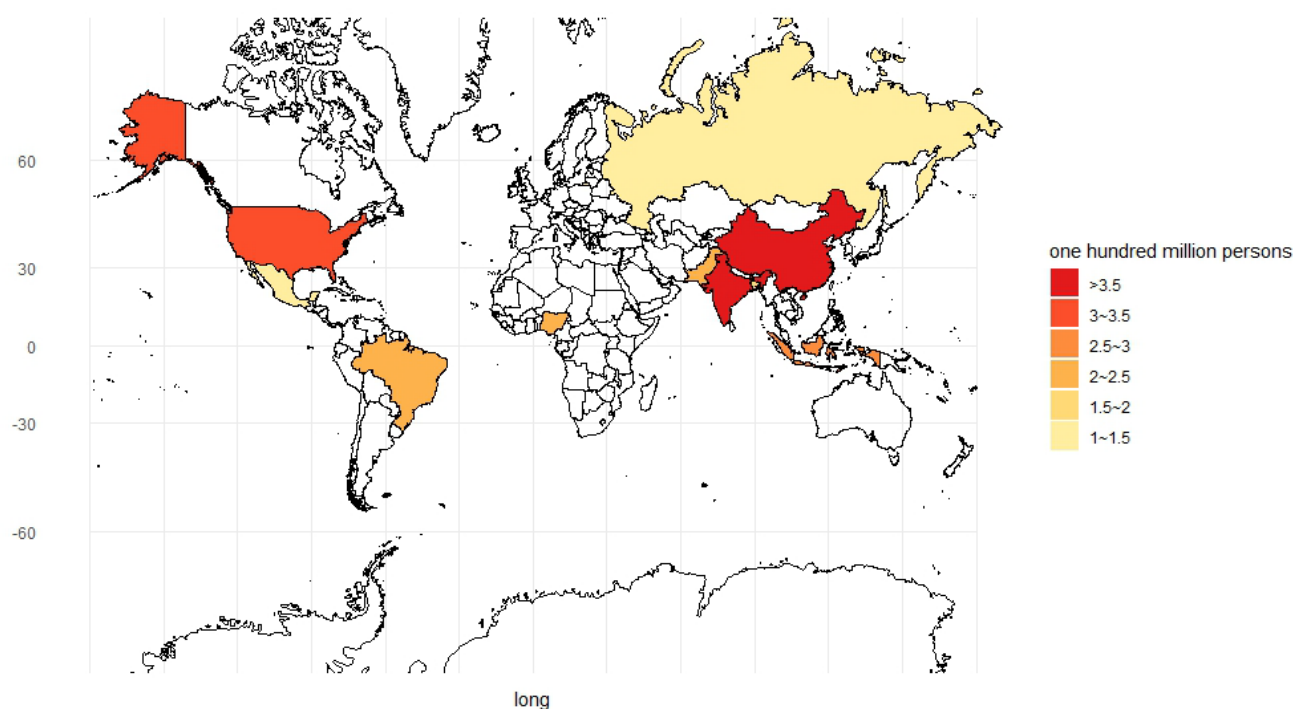


图 12 人口数量前 10 名国家地理图

由图 1.3.1 可知，人口数量前 10 名的国家分散于各个大洲。而以人口稠密程度为标志，印度无疑是人口最稠密地区，其次为中国。同时，印度尼西亚的人口稠密度也相对较高。

5.问题三的解决方案

5.1 数据预处理

基于已知数据，我们选择了人口数量具有相同量级的国家：巴西、俄罗斯、日本，并对其进行类比分析。在本题的解决过程中，我们主要使用 Excel 对原数据进行分组、聚合等处理，以得到三个国家不同年龄段分布的相关数据，再通过 R 语言进行可视化展示，并结合三个国家的国情分析各国不同年龄段人口的分布原因。

5.2.数据可视化展示与原因分析

5.2.1 人口分布直方图

通过对已知数据进行分类、筛选和简单计算后，我们得出 2021 年三个国家不同年龄段的人口分布直方图。如下图所示。

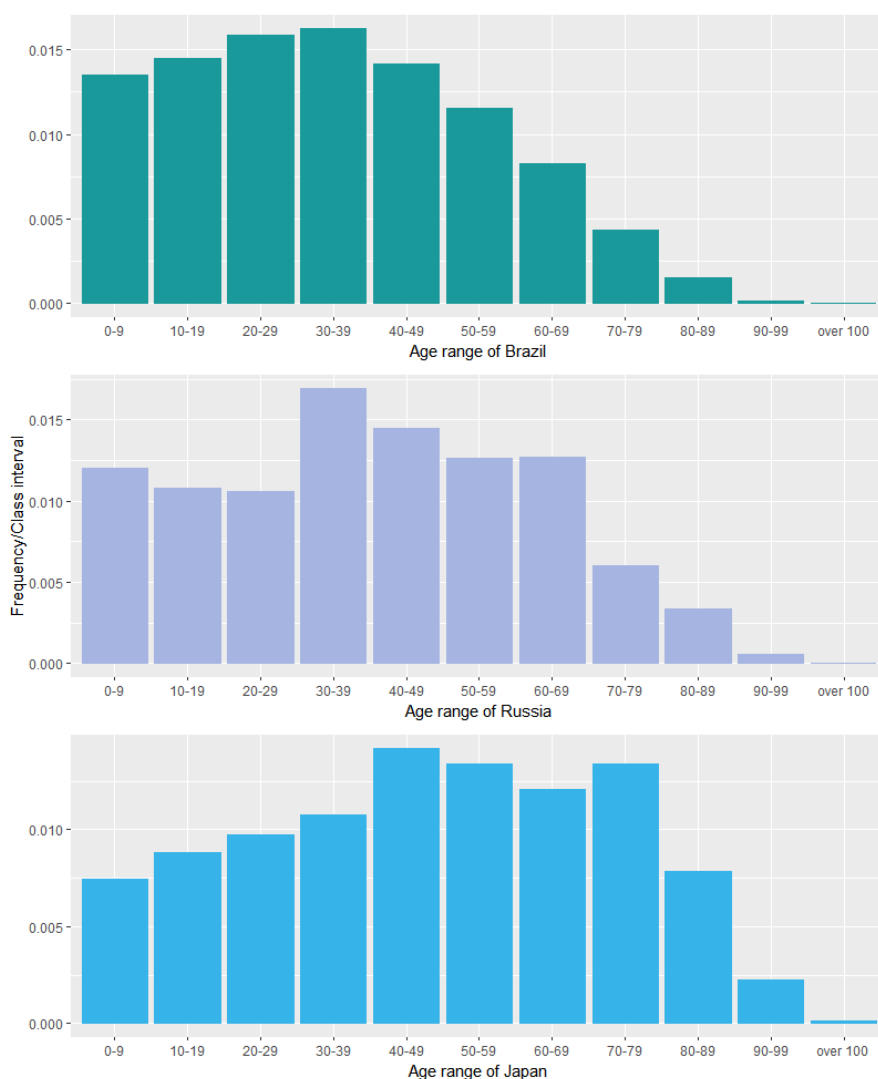


图 13 2021 年人口分布直方图

据图所示，我们容易看出 2021 年三个国家人口年龄分布具有不同的集中趋势。其中，巴西人口主要分布在 40 岁之前，俄罗斯人口主要分布在 30 至 70 岁之间，日本人口主要分布在 40 至 80 岁之间。同时可以看出，日本 60 岁以上的人口占总人口的比例相较巴西和俄罗斯明显偏大。相似的是，三个国家 100 岁以上人口占比都非常小。

5.2.2 人口堆叠面积图

为了进一步了解造成该现象的历史原因，我们结合已知数据对过去 71 年不同年龄段的人口分布进行了数据分析和可视化展示。通过对数据进行分类和预处理，我们得到如下所示的人口堆叠面积图。

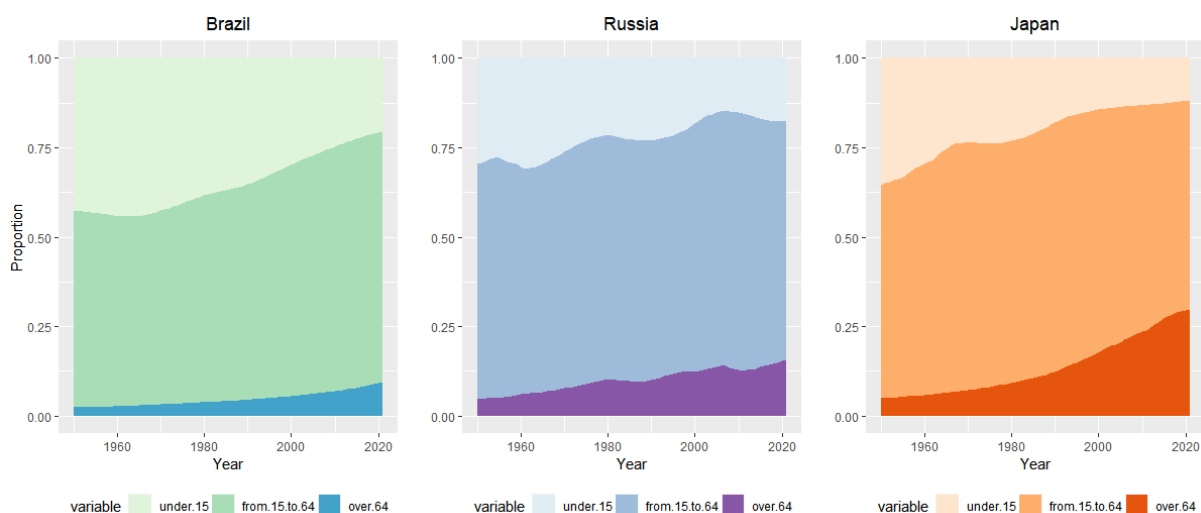


图 14 人口堆叠面积图

据图所示，在过去 71 年间，不同国家的人口年龄结构变化呈现相似的趋势。巴西、俄罗斯和日本 64 岁以上老人占总人口比重均呈上升趋势，其中，日本攀升最快；在三个国家中，15 岁以下儿童占总人口比重均有不同程度的下降。

5.2.3 人口年龄比率折线图

为了更加直观地比较三个国家老年人口和儿童人口比例的变化，我们对两个年龄段：大于 64 岁和小于 15 岁的人口占比变化数据进行处理，绘制了如下折线图。

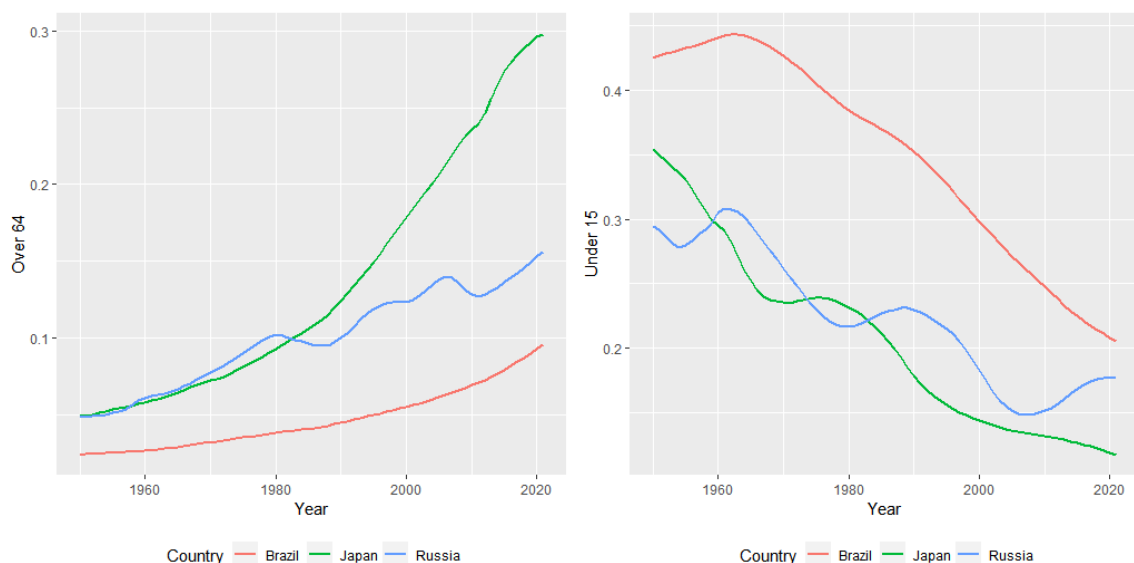


图 15 人口年龄比率折线图

据图所示，日本 64 岁以上老年人占比涨幅最大，在 2021 年的占比大约是俄罗斯老年人口占比的两倍。同时，巴西 15 岁以下儿童占比始终是三个国家占比中的最大的一项。俄罗斯的 64 岁以上老年人口占比变化和 15 岁以下儿童人口占比变化的波动均较大。

5.2.4 年龄分布原因解释

日本社会之所以呈现老龄化，一是日本经济发展的长期停滞诱发了“低欲望”社会的出现，因此年轻人开始“不婚化”，进而就是“少子化”[2]。毕竟，结婚人数的减少与出生率的下降是呈正相关关系；二是由于生活和医疗水平的提升，加之日本独特饮食文化的影响，所以，日本老年人更容易长寿[3]。在这两个因素的共同作用下，催生了日

本社会老龄化的出现。

俄罗斯人口也是年轻人偏少，原因同样是年轻人生育意愿的降低。由于俄罗斯经济发展存在诸多问题，加之寒冷气候的影响，俄罗斯人的生育意愿也很低。虽然也是低生育率，但是相比较日本，俄罗斯老龄化程度并不是太严重，主要原因应该是俄罗斯男性寿命偏短。生活在寒冷地区，新陈代谢变缓，俄罗斯人虽然具有长寿的客观条件，不过，长期、大量酗酒导致俄罗斯男性容易过早死亡，这就拉低了俄罗斯人寿命的平均值，所以，也就没有出现日本社会那样的老龄化现象^[4]。

在日本、俄罗斯和巴西这三个国家中，巴西人口最多，超过两亿。虽然巴西人口出生率也在逐年降低，不过，巴西的人口出生率始终保持在高位运行，所以，现在的巴西人口结构较为合理，即年轻人居多，老龄化程度尚不突出。相比较日本、俄罗斯两个高纬度国家，巴西因为气候炎热，所以国内的长寿者相对偏少。这在一定程度上也减缓了巴西老龄化社会的降临^[5]。

5.问题四的解决方案

5.1 理论准备

在进行人口预测时，由于不同国家的人口增长处于的历史阶段、以及预测时间节点的要求，要求我们选用不同的预测方法，而不同国家的人口增长的历史时期，可以分为以下几个部分：

1.初期：这个时期人口增长缓慢，主要原因是高婴儿死亡率和疾病的流行。这个时期通常是一个国家的历史上的早期时期。

2.加速期：这个时期人口增长开始加速，主要原因是医学技术的进步和卫生条件的改善。这个时期通常是一个国家工业化和城市化的时期。

3.减速期：这个时期人口增长开始减缓，主要原因是生育率下降和老龄化。这个时期通常是一个国家经济发展和社会进步的时期。

4.稳定期：这个时期人口增长趋于稳定，主要原因是生育率和死亡率趋于平衡。这个时期通常是一个国家经济和社会发展到了一个较高水平的时期。（当然在这一时期也有可能人口下降）

而在经过相关查询后，我们可以知道，中国已经到了第 3 个时期，在 2022 年甚至出现了人口负增长，而对于印度人口以及世界人口，其依旧处于稳步增长阶段，据联合国估计,到 2030 年时,印度的人口总数很可能突破 15 亿,而到 2060 年,印度的人口总数更是会达到 16.5 亿的峰值,随后开始下降。与此同时，预计全球人口数量将在 2050 年以前达到 97 亿，并于之后几十年内持续增长，最终达到 115 亿的高峰。

而在查阅相关资料后^[1],笔者以为，logistics 函数适合正在逐步增长期的人口预测，其主要公式为：

$$p(t) = \frac{p_m}{1 + (\frac{p_m}{p_0} - 1)e^{-l(t-t_0)}}$$

其中： p_m 为最大人口承载， p_0 为起始人口， $p(t)$ 为某一时间的人口， l 为某一固定人口增长率， t 为某个时间点， t_0 为起初时间点。

而由于所处的历史阶段与前二者均为不同，故而该公式不可行。同时在试用时间序列预测、灰色模型预测以及多种预测方法之后，贝叶斯神经网络曲线拟合效果较好，故而采用 matlab 对其进行神经网络预测。

5.2 参数选择以及模型构建

5.2.1 世界人口部分

由联合国数据以及原始数据，易知：

$$p_m = 11.499322e + 09, \frac{p_m}{p_0} = \frac{11}{2.5}, t_0 = 1950$$

得到用于曲线拟合的公式：

$$p(t) = \frac{11.499322e + 09}{1 + (\frac{11}{2.5} - 1)e^{-l(t-1950)}}$$

最终拟合得到：

系数(置信边界为 95%)：

$$l = 0.02731 \text{ (0.02703, 0.02759)}$$

表 7 拟合优度

数据指标	数值
SSE	1.323e+18
R 方	0.9935
调整 R 方	0.9935
RMSE	1.355e+08

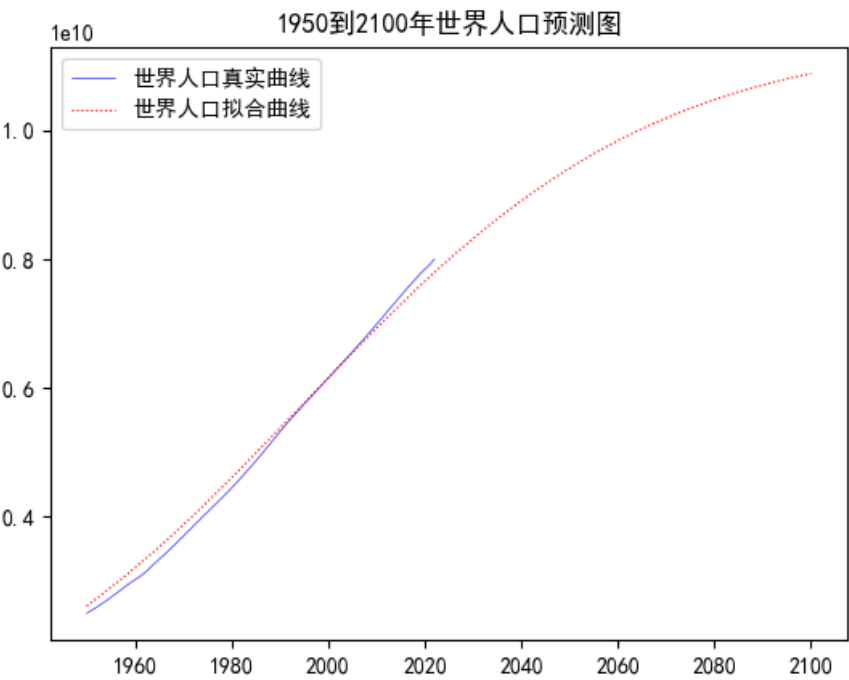


图 16 1950 到 2100 年世界人口预测图

之后，运用附录 4 的程序，绘制上图。

由数据易知，调整 R 方为 0.9935，曲线拟合效果较好，而从曲线来看也是如此，而从现在到 2100 年的数据放于“世界人口预测.xls”中，2100 年世界人口大致为 10883895846 人，约 109 亿。

5.2.2 印度人口部分

由联合国数据以及原始数据，易知：

$$p_m = 1.66e + 09, \frac{p_m}{p_0} = \frac{16.6}{3.57}, t_0 = 1950$$

得到用于曲线拟合的公式：

$$p(t) = \frac{1.66e + 09}{1 + (\frac{16.6}{3.57} - 1)e^{-l(t-1950)}}$$

最终拟合得到：

系数(置信边界为 95%):

$$l = 0.03745 \text{ (0.0367,0.0382)}$$

表 8 拟合优度

数据指标	数值
SSE	1.584e+17
R 方	0.9804
调整 R 方	0.9804
RMSE	4.69e+07

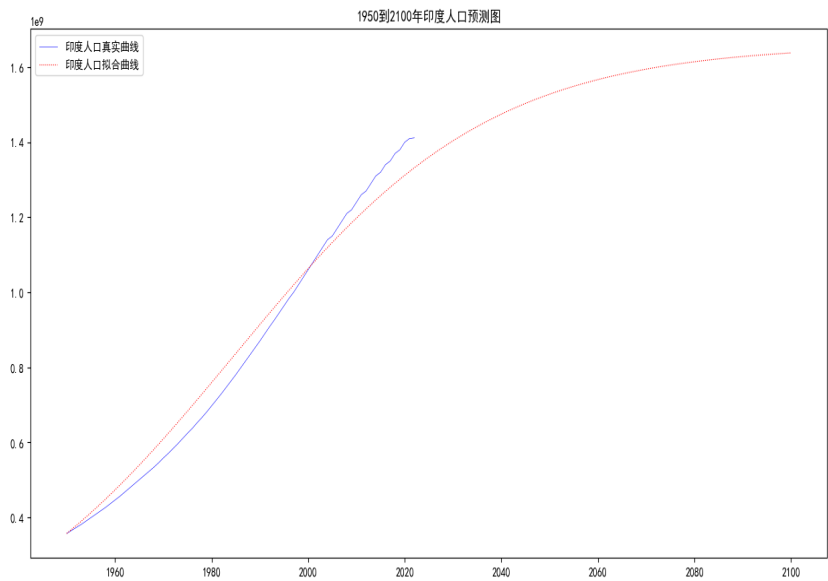


图 17 1950 到 2100 年印度人口预测图

之后，运用附录 5 的程序，绘制上图。

由数据易知，调整 R 方为 0.9804，曲线拟合效果较好。而从曲线来看，则感觉偏差显大，从现在到 2100 年的数据放于“印度人口预测.xls”中，2100 年世界人口大致为 1638272336 人，约 16 亿。

5.2.3 中国人口部分

该部分，运用计算机模拟进行，经过相关文献记载[6],我们可以得到如下生育率表格，低方案根据 2023 年的数据进行一定的修正，改为 1.03，本模型先作出如下假设：

- (1) 假如每年出生人口的男女比例为 1:1。
- (2) 假如之后每一年的生育率都保持在该方案的总和生育率下。

而具体的生育率统计表格如下^[6]:

表 9 不同方案生育率统计表格

	不采取措施	采取措施
一孩生育率	0.55	0.70
二孩生育率	0.33	0.50
三孩生育率	0.15	0.25
合计	1.03	1.45

本模拟程序的思路是这样的（具体的放在附录 7 与附录 8 中）：

首先先找到各个年龄段的性别、人口以及妇女分年龄段生育的数据，之后创建一个人口池，算出每个部分各个年龄段的人的岁数，然后循环往复，以此对各个年龄段的出生人口以及死亡人口进行预测，概率使用二项分布，概率为各个年龄段人口的死亡率与出生率，最后一年一年地进行模拟，得到最终的中国人口预测数据，预测可视化结果如下（而具体措施都保存在文件夹中）：

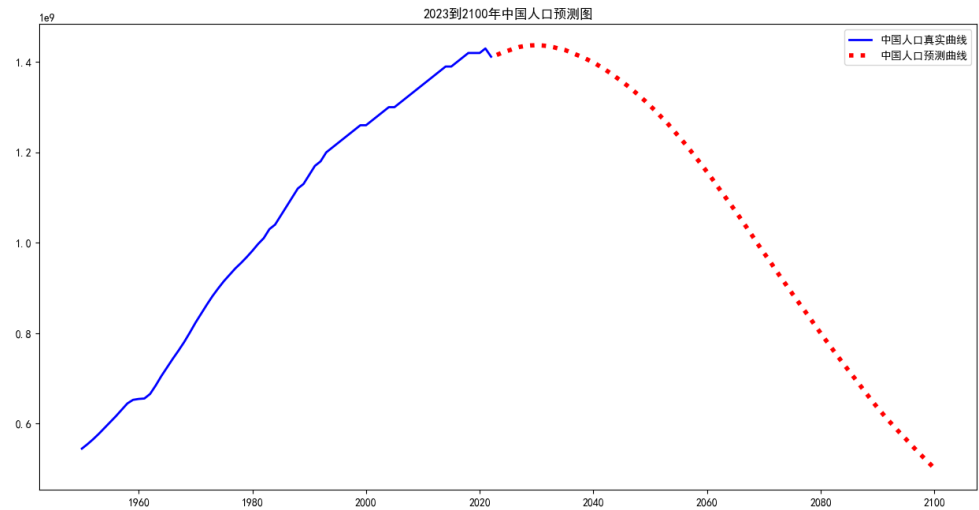


图 18 2023 到 2100 中国人口预测图（不采取措施情况）

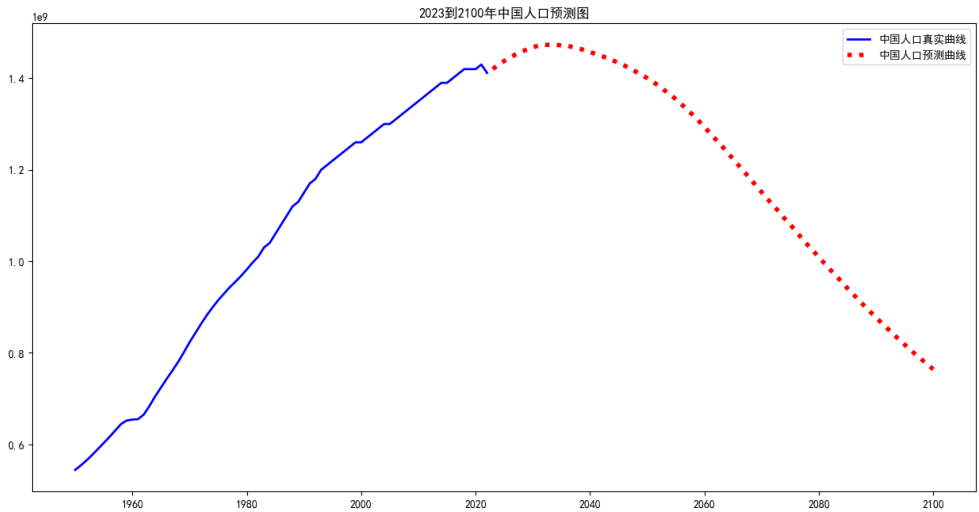


图 19 2023 到 2100 中国人口预测图（采取措施情况）

由图可见，加入中国人口不采取措施进行补救，那么到 2050 年，人口大致降为 13

亿，而到了 2100 年左右，将降为 5 亿左右。

而如果采取措施补救，2050 年人口为 14 亿，而 2100 将降为 8 亿。

6.总结以及建议

对于问题一：我们可以看到，世界人口在不断增长，这需要引起我们的注意，需要加强各国的人口管控，同时的话，协调好发展以及环境保护之间的关系，提高环境承载力。与此同时，我们还不难发现，世界的主要人口增长集中于中东以及非洲地区，而发达国家的人口则保持稳定甚至是负增长，地区人口增长的不协调现象尤为显著，发展不平衡是导致这一现象的重要原因，这就要求我们在国际合作的同时彼此谅解，就各个国家不同的国情采取对应的人口政策。

对于问题二：中国和印度作为世界第一、第二的人口大国，应该加强对人口政策的研究和制定，注意控制人口增长速度，加强宣传教育，提高人民对计划生育的认识，推动社会进步，让人口合理增长。而作为人口超过 3 亿，世界第三人口的发达国家，美国应该加强对老龄化社会的应对，加大对儿童、老人等特殊人口的关爱力度，提高社会保障水平。对于作为东南亚最大的国家的印度尼西亚，应加强基础设施建设，提高教育和医疗水平，为人民创造更好的生活条件。巴基斯坦近年来人口增长迅猛，所以巴基斯坦应该加强对教育的投资，提高妇女和儿童的保健水平，加强对生育率的限制。而其他人口众多的国家，也需要加大对社会基础设施与人口老龄化的关注程度。

对于全球人口数量后十名的国家，因为环境、地理、国土面积、社会意识等因素的影响，不能过度追求人口的数量增加，应该要注重对基础设施和教育的投资，提升人民生活水平，同时注重环境和资源的保护，协调好人与自然的关系。

对于问题三：人口出生率 and 经济发展程度呈反比，在经济发展还不够充分的国家，出生率通常偏高，如巴西。反之，则是出生率降低，如日本和俄罗斯。出生率偏低直接导致人口的老龄化。所以，发达国家通常是老龄化社会。这个情况在所选取的上述三个国家中十分明显。人口是战略资源，国家的发展需要庞大且众多的人口作为支撑。人口老龄化会对经济发展产生诸多不利影响。所以，我们国家需要制定提高人口出生率的相关政策，以便为国家的可持续发展打下人口基础。

对于问题四：世界以及印度的人口依旧在不断增长，需要世界各国以及印度自身加强管控。在未来，我国人口进入负增长阶段，政府应当通过社会配套保障措施，推动机器人进入养老领域等应对人口老龄化问题，同时推行政策、建立良好的社会保障制度，使得青年人能生育、敢生育。

7.参考文献

- [1]王沛林.四类预测人口方法的对比及 Logistic 人口生长模型的改进[J].保定学院学报,2022,35(03):87-98.DOI:10.13747/j.cnki.bdxxyxb.2022.03.014.
- [2]宋金文.日本少子化的现状、对策与困境[J].社会政策研究,2022,No.29(04):3-24.DOI:10.19506/j.cnki.cn10-1428/d.2022.04.003.
- [3]冯维希.日本重度老龄化社会老年人贫困问题研究[D].对外经济贸易大学,2022.DOI:10.27015/d.cnki.gdwju.2022.000349.
- [4]朱香玉.俄罗斯人口形势现状及发展趋势研究[J].西伯利亚研究,2020,47(01):47-57+118.
- [5]胡可.近十年来金砖国家人口地理特征比较研究[D].山西师范大学,2012.
- [6]陈卫.中国人口负增长与老龄化趋势预测[J].社会科学辑刊,2022,No.262(05):133-144.

8.附录

附录 1：统计各国人口增长率及其增长人口.py

```
from openpyxl import load_workbook

wb=load_workbook(r"C:\Users\Administrator\Desktop\vs 工作界面\各个国家人口数据.xlsx")
ws=wb.active
ws.cell(1,4,"人口增长率")
ws.cell(1,5,"人口增长")

for i in range(2,15987,72):    ##就行数进行遍历，同时 72 是从一个国家跳到另外一个国家的步数，表示 1950-2022 这几年
    ws.cell(i,4,0)
    ws.cell(i,5,0)
    for j in range(1,72):
        ws.cell(i+j,5,ws.cell(i+j,3).value-ws.cell(i+j-1,3).value)
        ws.cell(i+j,4,ws.cell(i+j,5).value/ws.cell(i+j,3).value)

wb.save(r"C:\Users\Administrator\Desktop\vs 工作界面\各个国家人口数据(已处理).xlsx")
```

附录 2：统计平均人口增长率及其绝对数.py

```
from openpyxl import load_workbook
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

wb=load_workbook(r"C:\Users\Administrator\Desktop\vs 工作界面\各个国家人口数据（已处理）.xlsx")
ws=wb.active

nation_growth=dict()
nation_num=dict()
sub_year=range(2011,2022)

for i in range(1,16057,1):    ##逐行读入，只统计 2011 年到 2022 年的，同时的话用字典判别不同国家
    year=ws.cell(i,2).value
```

```

nation=ws.cell(i,1).value
if(nation not in nation_growth.keys()):
    nation_growth[nation]=0
    nation_num[nation]=0
if(year in sub_year):
    nation_growth[nation]+=ws.cell(i,4).value
    nation_num[nation]+=ws.cell(i,5).value

for i in nation_growth.keys():
    nation_growth[i]/=11
    nation_num[i]/=11
##对于人口增长率以及人口增长进行排序
res1=sorted(nation_growth.items(),key=lambda d:d[1],reverse=True)
res2=sorted(nation_num.items(),key=lambda d:d[1],reverse=True)

x,y=[],[]
print("人口增长率前 10:")
for i in range(0,10):
    print(res1[i][0])
    x.append(res1[i][0])
    y.append(round(res1[i][1],4))
data=pd.DataFrame(data=np.transpose(np.array([x,y])),columns=["国家","人口增长率前 10"])
data.to_excel("人口增长率前 10 表.xlsx")

x,y=[],[]
print("人口增长前 10:")
for i in range(0,10):
    print(res2[i][0])
    x.append(res2[i][0])
    y.append(math.floor(res2[i][1]))
data=pd.DataFrame(data=np.transpose(np.array([x,y])),columns=["国家","人口增长前 10"])
data.to_excel("人口增长前 10 表.xlsx")

x,y=[],[]
print("人口增长率倒数后 10")
for i in reversed(res1[-10:]):
    print(i[0])
    x.append(i[0])
    y.append(round(-i[1],4))
data=pd.DataFrame(data=np.transpose(np.array([x,y])),columns=["国家","人口增长率倒数后 10"])
data.to_excel("人口增长率倒数后 10 表.xlsx")

x,y=[],[]

```

```

print("人口增长倒数后 10")
for i in reversed(res2[-10:]):
    print(i[0])
    x.append(i[0])
    y.append(math.floor(-i[1]))
data=pd.DataFrame(data=np.transpose(np.array([x,y])),columns=["国家","人口增长倒数后 10"])
data.to_excel("人口增长倒数后 10 表.xlsx")

```

附录 3：绘制总人口曲线变化图.py

```

import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

data=pd.read_excel(r"C:\Users\Administrator\Desktop\vs 工作界面\世界总人口变化.xls",usecols=["Year","Population"])
fig=plt.figure(figsize=(4, 4), dpi=150)
plt.plot(data["Year"],data["Population"],lw=3,ls='-',c='b',alpha=0.5,label="人口变化曲线")
plt.title(label="1950 到 2021 世界人口变化图")
for i in range(0,71,10):
    plt.text(data["Year"][i]-2,data["Population"][i]*0.90,s="{:} 亿".format(data["Year"][i],round(data["Population"][i]/(100000000),2)),fontsize=6)
plt.legend()
plt.show()

```

附录 4：全球人口画图以及相关数据导出.py

```

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

year=np.arange(1950,2101)
pop_predic=11.499322e+09/(1+(11/2.5-1)*np.e**(-0.02731*(year-1950)))    ##相应的预测曲线
pop_ori=pd.read_excel(r"C:\Users\Administrator\Desktop\vs 工作界面\世界总人口变化.xls",usecols=["Population"])  ##读入原始数据
pop_ori=np.array(pop_ori)

plt.plot(np.arange(1950,2023),pop_ori,lw=0.4,color="b",linestyle="-",label="世界人口真实曲线")

```



```
plt.plot(year,pop_predic,lw=0.8,color="r",linestyle=":",label="世界人口拟合曲线")
plt.title("1950 到 2100 年世界人口预测图")
plt.legend()
plt.show()

data=pd.DataFrame(data=[year,pop_ori,pop_predic],index=["年份","原始数据人口","预测人口"])
data.to_excel("世界人口预测.xls")
```

附录 5：印度人口画图以及相关数据导出.py

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

year=np.arange(1950,2101)
pop_predic=1.66e+09/(1+(16.6/3.57-1)*np.e**(-0.03745*(year-1950)))    ##算出预测值
pop_ori=pd.read_excel(r"C:\Users\Administrator\Desktop\vs 工作界面\印度总人口变化.xls",usecols=["Population"])    ##读入已知值
pop_ori=np.array(pop_ori)

plt.plot(np.arange(1950,2023),pop_ori,lw=0.4,color="b",linestyle="-",label="印度人口真实曲线")
plt.plot(year,pop_predic,lw=0.8,color="r",linestyle=":",label="印度人口拟合曲线")
plt.title("1950 到 2100 年印度人口预测图")
plt.legend()
plt.show()

data=pd.DataFrame(data=[year,pop_ori,pop_predic],index=["年份","原始数据人口","预测人口"])
data.to_excel("印度人口预测.xls")
```

附件 6：中国人口画图.py

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

year=np.arange(1950,2101)
pop_ori=pd.read_excel(r"C:\Users\Administrator\Desktop\vs 工作界面\中国人口预
```

```

测.xlsx",usecols=["原始数据人口"])
pop_ori.dropna(how="any",inplace=True)
pop_ori=np.array(pop_ori)

pop_predic=pd.read_excel(r"C:\Users\Administrator\Desktop\vs 工作界面\中国人口预测.xlsx",usecols=["预测人口"])
pop_predic=np.array(pop_predic)

plt.plot(np.arange(1950,2023),pop_ori,lw=2,color="b",linestyle="-",label="中国人口真实曲线")
plt.plot(year,pop_predic,lw=4,color="r",linestyle=":",label="中国人口拟合曲线")
plt.title("1950 到 2100 年中国人口预测图")
plt.legend()
plt.show()

```

附录 7：中国人口预测（无措施） py

```

import pandas as pd
import numpy as np
import copy
import sys
# import multiprocessing as Pool

total_pop_cur=1411750000
CURRENT_YEAR=2022 ##现在的年份
chi_mean=float(1.08)    ##设置中国总和生育率
sd=10000    ##代表将其变成万
sex_bir_ratio=0.5    ##代表男女比例

age_sex = pd.read_csv("./age_sex.csv", header=None,

names=["age","total","male","female","total_prop","male_prop","female_prop","sex_ratio"])
##读入相关的性别数据
#不知道的人口信息用第六次人口普查的代替
male_prop = age_sex['male']/sum(age_sex['total'])
female_prop = age_sex['female']/sum(age_sex['total'])    ##计算对应不同年龄段人口占总人口的比例

sex_n = pd.DataFrame({"male" : male_prop*total_pop_cur/sd,
                      "female" : female_prop*total_pop_cur/sd})
sex_n = sex_n.astype({"male":"int", "female":"int"})    ##对应的男女人口

#全国分年龄、性别的死亡人口状况
age_death = pd.read_csv("./age_death_per_year.csv", header=None,

```

```

names=["age","total","male","female","total_death","male_death","female_death",
        "total_dt","male_dt","female_dt"])
death_rate = age_death[{"male_dt","female_dt"}]/1000

#全国育龄妇女分年龄、孩次的生育状况
age_birth = pd.read_csv("./birth.csv", header = None,

names=["age","total_women","total_birth","birth_rate","firstBirth_number","firstBirth_rate",

"secondBirth_number","secondBirth_rate","thirdBirth_number","thirdBirth_rate"])

birth_rate_adj = pd.DataFrame({'age' : age_birth['age'],
                                "firstBirth_rate"
                                :
age_birth["firstBirth_rate"]/sum(age_birth["firstBirth_rate"])}))

class Person(object):    ##创建模拟类
    def __init__(self, sex,bir_yea,childNO):
        self.sex=sex    #1 是男性， 2 是女性
        self.bir_yea=bir_yea
        self.childNO=childNO    ##代表有无小孩
    def age(self,Year): ##代表返回其年龄
        return Year-self.bir_yea
    def status(self, Year):
        if self.sex==1: sexname = "male_dt"
        if self.sex==2: sexname = "female_dt"
        if self.age(Year)>100:
            dr=death_rate.iloc[100][sexname]
        else:
            dr=death_rate.iloc[self.age(Year)][sexname]
        return np.random.binomial(1,dr) #1 代表死了， 0 代表活着
    def birth(self,Year,chi_mean):
        if self.age(Year)<15 or self.age(Year)>49 or self.sex==1 or self.status(Year)==1:    ##
        代表着不能生育的人
            return None
        else:
            br = birth_rate_adj.loc[birth_rate_adj['age']==self.age(Year)]['firstBirth_rate']
            bn = np.random.binomial(1,br*chi_mean)    ##设定其为二项分布，代表生育
            与否
            if bn==1:
                self.childNO+=bn
                cc=Person(np.random.binomial(1,sex_bir_ratio)+1,Year,0)
                return cc
            return None

# initiate a pool of starting population

```

```

def initPop():
    p_pool = []  ##人口池
    for iage, irow in sex_n.iterrows():  ##包含该行的信息以及对应的索引
        for ii in range(irow['female']):  ##计算一下人口的出生年份
            p_pool.append(Person(2,CURRENT_YEAR-iaage,0))
        for jj in range(irow['male']):
            p_pool.append(Person(1,CURRENT_YEAR-iaage,0))
    return p_pool

if __name__ == '__main__':
    p_pool = initPop()
    pop_log = {}
    year=2023
    while True:
        print(year)
        for p in p_pool:
            pbirth = p.birth(year, chi_mean)
            if pbirth is not None:  ##代表生育
                p_pool.append(pbirth)
            if p.status(year)==1:
                p_pool.remove(p)
        pop_log[year]=copy.deepcopy(p_pool) # deep copy.
        if year==2100:
            break
        year=year+1
    out = open("中国人口预测.txt","w")
    out.write("Year\tMale_pop\tFemale_pop\n")
    for k,plist in pop_log.items():
        male_no = len([p for p in plist if p.sex == 1])
        female_no = len([p for p in plist if p.sex == 2])
        out.write(str(k) + "\t" + str(male_no) + "\t" + str(female_no) + "\n")
    out.close()

```

附录 8：中国人口预测（有措施） py

```

import pandas as pd
import numpy as np
import copy
import sys
# import multiprocessing as Pool

total_pop_cur=1411750000
CURRENT_YEAR=2022 ##现在的年份
chi_mean=float(1.45)  ##设置中国总和生育率
sd=10000  ##代表将其变成万

```

```

sex_bir_ratio=0.5      ##代表男女比例

age_sex = pd.read_csv("./age_sex.csv", header=None,

names=["age","total","male","female","total_prop","male_prop","female_prop","sex_ratio"])
##读入相关的性别数据
#不知道的人口信息用第六次人口普查的代替
male_prop = age_sex['male']/sum(age_sex['total'])
female_prop = age_sex['female']/sum(age_sex['total'])      ##计算对应不同年龄段人口
                    占总人口的比例

sex_n = pd.DataFrame({"male" : male_prop*total_pop_cur/sd,
                      "female" : female_prop*total_pop_cur/sd})
sex_n = sex_n.astype({"male":"int", "female":"int"})      ##对应的男女人口

#全国分年龄、性别的死亡人口状况
age_death = pd.read_csv("./age_death_per_year.csv", header=None,
                        names=["age","total","male","female","total_death","male_death","female_death",
                              "total_dt","male_dt","female_dt"])
death_rate = age_death[{"male_dt","female_dt"}]/1000

#全国育龄妇女分年龄、孩次的生育状况
age_birth = pd.read_csv("./birth.csv", header = None,

names=["age","total_women","total_birth","birth_rate","firstBirth_number","firstBirth_rate",

"secondBirth_number","secondBirth_rate","thirdBirth_number","thirdBirth_rate"])

birth_rate_adj = pd.DataFrame({'age' : age_birth['age'],
                              "firstBirth_rate"
                              :
age_birth["firstBirth_rate"]/sum(age_birth["firstBirth_rate"])))

class Person(object):    ##创建模拟类
    def __init__(self, sex,bir_yea,childNO):
        self.sex=sex    #1 是男性，2 是女性
        self.bir_yea=bir_yea
        self.childNO=childNO      ##代表有无小孩
    def age(self,Year): ##代表返回其年龄
        return Year-self.bir_yea
    def status(self, Year):
        if self.sex==1: sexname = "male_dt"
        if self.sex==2: sexname = "female_dt"
        if self.age(Year)>100:
            dr=death_rate.iloc[100][sexname]
        else:

```

```

        dr=death_rate.iloc[self.age(Year)][sexname]
        return np.random.binomial(1,dr) #1 代表死了， 0 代表活着
    def birth(self,Year,chi_mean):
        if self.age(Year)<15 or self.age(Year)>49 or self.sex==1 or self.status(Year)==1:
##代表着不能生育的人
            return None
        else:
            br
            =
            birth_rate_adj.loc[birth_rate_adj['age']==self.age(Year)]['firstBirth_rate']
            bn = np.random.binomial(1,br*chi_mean)    ##设定其为二项分布，代表
            生育与否
            if bn==1:
                self.childNO+=bn
                cc=Person(np.random.binomial(1,sex_bir_ratio)+1,Year,0)
                return cc
            return None

# initiate a pool of starting population
def initPop():
    p_pool = []    ##人口池
    for iage, irow in sex_n.iterrows():    ##包含该行的信息以及对应的索引
        for ii in range(irow['female']):    ##计算一下人口的出生年份
            p_pool.append(Person(2,CURRENT_YEAR-iage,0))
        for jj in range(irow['male']):
            p_pool.append(Person(1,CURRENT_YEAR-iage,0))
    return p_pool

if __name__ == '__main__':
    p_pool = initPop()
    pop_log = {}
    year=2023
    while True:
        print(year)
        for p in p_pool:
            pbirth = p.birth(year, chi_mean)
            if pbirth is not None:    ##代表生育
                p_pool.append(pbirth)
            if p.status(year)==1:
                p_pool.remove(p)
        pop_log[year]=copy.deepcopy(p_pool) # deep copy.
        if year==2100:
            break
        year=year+1
    out = open("中国人口预测.txt","w")

```

```

out.write("Year\tMale_pop\tFemale_pop\n")
for k,plist in pop_log.items():
    male_no = len([p for p in plist if p.sex == 1])
    female_no = len([p for p in plist if p.sex == 2])
    out.write(str(k) + "\t" + str(male_no) + "\t" + str(female_no) + "\n")
out.close()

```

附录 9：问题三部分画图 R 代码

人口堆叠柱状图

```

library(ggplot2)
df<-read.csv("Japan2.csv")
df1<-read.csv("Brazil2.csv")
df2<-read.csv("Russia2.csv")
df_melt<-melt(df,id="Year")
df_melt1<-melt(df1,id="Year")
df_melt2<-melt(df2,id="Year")
a<-ggplot(df_melt, aes(x=Year, y=value, group=variable)) +
  geom_area(stat="identity",position="stack",aes(fill=variable)) +
  labs(x = "Year",y = "")+
  scale_fill_brewer(palette = 7)+
  theme(legend.position = "bottom",
        plot.title = element_text(hjust = 0.5))+
  ggtitle("Japan")
b<-ggplot(df_melt1, aes(x=Year, y=value, group=variable)) +
  geom_area(stat="identity",position="stack",aes(fill=variable)) +
  labs(x = "Year",y = "Proportion")+
  scale_fill_brewer(palette = 4)+
  theme(legend.position = "bottom",plot.title = element_text(hjust = 0.5))+
  ggtitle("Brazil")
c<-ggplot(df_melt2, aes(x=Year, y=value, group=variable)) +
  geom_area(stat="identity",position="stack",aes(fill=variable)) +
  labs(x = "Year",y = "")+
  scale_fill_brewer(palette = 3)+
  theme(legend.position = "bottom",plot.title = element_text(hjust = 0.5))+
  ggtitle("Russia")
Rmisc::multiplot(b,c,a,cols=3)

```

折线图

```

data<-read.csv("rate1.csv")
data1<-read.csv("rate.csv")
a<-ggplot(data,aes(x=Year,y=Aging.rate,color=Country))+
  geom_line(size=0.8)+ylab("Under 15")+
  theme(legend.position = "bottom")

```

```

b<-ggplot(data1,aes(x=Year,y=Aging.rate,color=Country))+
  geom_line(size=0.8)+ylab("Over 64")+
  theme(legend.position = "bottom")
Rmisc::multiplot(b,a,cols=2)

```

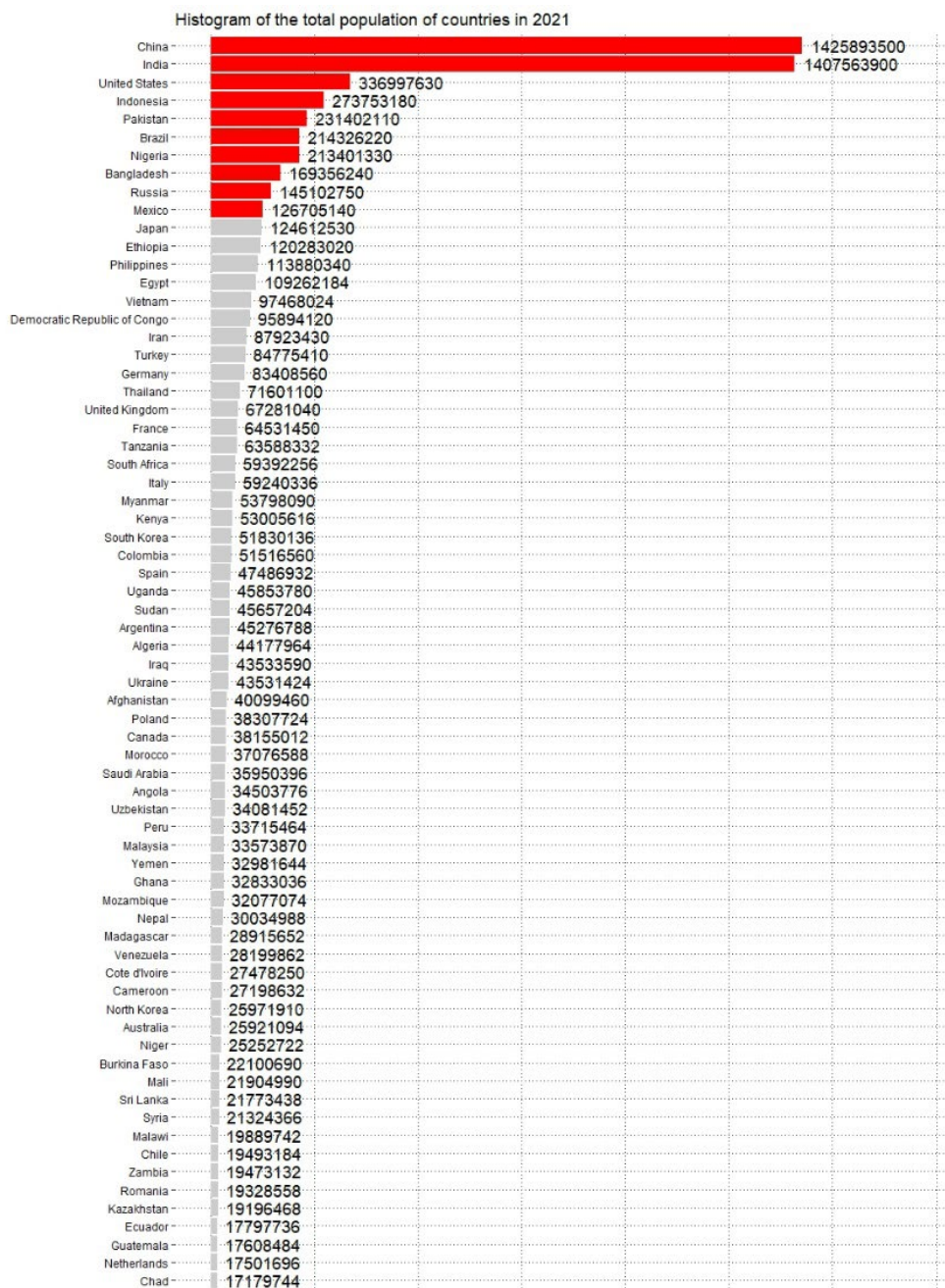
直方图

```

mydata<-read.csv("Brazil.csv")
mydata1<-read.csv("Russia1.csv")
mydata2<-read.csv("Japan1.csv")
a<-ggplot(mydata,aes(x=Age.interval,y=Population.rate))+
  geom_bar(stat="identity",color="#199999",fill="#199999")+
  xlab("Age range of Brazil")+
  ylab("")
b<-ggplot(mydata1,aes(x=Age.interval,y=Population.rate))+
  geom_bar(stat="identity",color="#A6B4E1",fill="#A6B4E1")+
  xlab("Age range of Russia")+
  ylab("Frequency/Class interval")
c<-ggplot(mydata2,aes(x=Age.interval,y=Population.rate))+
  geom_bar(stat="identity",color="#36B4E9",fill="#36B4E9")+
  xlab("Age range of Japan")+
  ylab("")
Rmisc::multiplot(a,b,c,line=3)

```


附录 10：各个国家人口数据排名



	Chad	17179744		
	Somalia	17065588		
	Senegal	16876726		
	Cambodia	16589031		
	Zimbabwe	15993525		
	Guinea	13531909		
	Rwanda	13461891		
	Benin	12996901		
	Burundi	12551215		
	Tunisia	12262949		
	Bolivia	12079474		
	Belgium	11611416		
	Haiti	11447575		
	Cuba	11256373		
	Jordan	11148288		
	Dominican Republic	11117873		
	South Sudan	10748278		
	Czechia	10510748		
	Sweden	10467095		
	Greece	10445368		
	Azerbaijan	10312992		
	Portugal	10290109		
	Honduras	10278346		
	Papua New Guinea	9949438		
	Tajikistan	9750078		
	Hungary	9709784		
	Belarus	9578172		
	United Arab Emirates	9365149		
	Austria	8922086		
	Israel	8900057		
	Switzerland	8691409		
	Togo	8644829		
	Sierra Leone	8420642		
	Laos	7425055		
	Serbia	7296771		
	Bulgaria	6885864		
	Nicaragua	6850546		
	Libya	6735280		
	Paraguay	6703802		
	Kyrgyzstan	6527742		
	Turkmenistan	6341850		
	El Salvador	6314165		
	Singapore	5941063		
	Denmark	5854246		
	Congo	5835814		
	Lebanon	5592626		
	Finland	5535982		
	Central African Republic	5457165		
	Slovakia	5447621		
	Norway	5403021		
	Liberia	5193422		
	Costa Rica	5153959		
	Palestine	5133393		
	New Zealand	5129730		
	Ireland	4986525		
	Mauritania	4614981		
	Oman	4520474		
	Panama	4351264		
	Kuwait	4250111		
	Croatia	4060139		
	Georgia	3757984		
	Eritrea	3620324		
	Uruguay	3426265		
	Mongolia	3347782		
	Bosnia and Herzegovina	3270948		
	Puerto Rico	3256030		
	Moldova	3061509		
	Albania	2854710		
	Jamaica	2827701		
	Armenia	2790971		
	Lithuania	2786652		
	Qatar	2688239		
	Gambia	2639922		
	Botswana	2588424		
	Namibia	2530150		
	Gabon	2341185		
	Lesotho	2281464		
	Slovenia	2119408		
	North Macedonia	2103329		
	Guinea-Bissau	2060730		
	Latvia	1873926		
	Kosovo	1662022		
	Equatorial Guinea	1634473		
	Trinidad and Tobago	1525671		
	Bahrain	1463266		
	Estonia	1328704		
	Timor	1320944		
	Mauritius	1298922		
	Cyprus	1244193		
	Eswatini	1192273		
	Djibouti	1105562		
	Reunion	966132		
	Fiji	924615		
	Comoros	821632		
	Guyana	804571		
	Bhutan	777500		
	Luxembourg	639330		
	Montenegro	627856		
	Suriname	612989		
	Cape Verde	587936		
	Western Sahara	565590		
	Malta	526751		
	Maldives	521469		
	Brunei	445382		

