北京航空航天大學
BEIHANG UNIVERSITY

2020 年数学建模

基于$SEIR$模型的新冠病毒建模与预测

姓名及学号　　朱英豪 18373722
姓名及学号　　任昌禹 18373718
姓名及学号　　胡鹏飞 18373059

摘要

本文针对由新冠病毒在全球肆虐而无法得到有效控制这一重大社会问题，建立了基于$SEIR$的传染病模型。针对干预前和干预后两个不同阶段，分别建立了模型进行分析，并结合已经控制住疫情的中国的数据，较为准确的预测了美国等国的未来感染人数。此外，本文创新性地根据患者在不同患病阶段具有不同的传染效果的特性，建立了优化后的$SEIR$模型，更为严谨地对各地区进行了预测。此外，我们对干预效果对疫情防治的效果进行了模拟分析，证明了“早发现、早隔离、严防严控的”的必要性。根据当前疫情局势，可知我国疫情基本得到控制，未来需主要警惕境外输入，以防止疫情的反弹。世界范围内，疫情仍处在上升阶段，但预计于 5 月中旬开始得到有效控制。本文的$SEIR$模型解决方案可适用于各种大规模传染病的分析与预测。

关键字：$SEIR$、传染病模型、新冠肺炎、微分方程

目录

1 介绍 5
1.1 问题提出 5
1.2 解决方案概述 5
2 符号说明 5
3 总体假设 6
4 干预前模型 6
4.1 本地参数 6
4.2 本地假设 6
4.3 模型建立 7
4.4 本地参数的确定 7
5 干预后模型 8
5.1 本地参数 8
5.2 本地假设 8
5.3 模型建立 8
5.4 本地参数的确定 9
6 优化模型 9
6.1 本地定义 10
6.2 本地参数 10
6.3 本地假设 10
6.4 模型建立 11
6.5 本地参数的确定 11
7 干预效果模拟分析 12
8 预测结果及分析 18
8.1 湖北、武汉 18

8.2 美国 19
8.3 英国 20
8.4 日本 21
8.5 累积和新增境外输入病例趋势 22
9 模型评价 22
9.1 优点 22
9.2 不足 23
9.3 未来工作 23
参考文献与数据来源 24
附录 25

1 介绍

1.1 问题提出

2019 年 12 月，湖北省武汉市陆续发现了多例不明原因的肺炎病例，2020 年 1 月末开始，疫情迅速发展，截止 2020 年 4 月末，我国累计确诊人数已达 84000 余例，世界范围内的累计确诊人数已逾 200 万例。当前，中国已基本上控制住了新冠病毒的传播，每日新增本土病例基本清零，而随着其他各国的感染人数的迅速攀升，在疫情局势极为严峻的当下，为疫情的发展进行建模预测分析是非常有必要的。本论文根据中国新冠病毒感染人数的变化以及部分其他国家的新冠病毒的相关数据，结合新冠病毒的传染特性建立$SEIR$传染病模型，分析全世界当前的疫情态势，对部分地区的疫情作出预测，以期为全球抗击这场疫情提出建设性意见。

1.2 解决方案概述

首先从疫情爆发初期，缺乏人为干预的较简单情况进行建模分析，将人群划分为健康者，潜伏者，病人，退出者，建立并求解微分方程，得到病毒传播的一般规律。然后，考虑人为干预后的相关因素，部分病人将被有效隔离与收治等情况，对模型进行修正，将病人进一步划分为隔离可控者、未隔离者以及自由带菌者，得到人为干预下病毒的传播规律。此外，我们建立了优化模型，进一步考虑实际情况中的不确定因素以及不同患病程度的病人相应传染病学性质的差异，将病人再细分为轻微者、严重者和重症患者，分别假设不同的治愈率和传染率，使模型更加接近真实情况。对于干预措施的有效程度的分析，通过数值模拟，并进行了控制变量法分析。最后，我们结合疫情的相关统计数据，使用 R 语言求解与分析，并将建模结果可视化，绘制出了病情预测曲线，对未来疫情的发展作出了估计。

2 符号说明

- S：健康者（易受感染者）在人群中的比例
- E：潜伏期者（尚未发病，但最终会发病）在人群中的比例
- I：病人在人群中的比例
- R：退出者（包含治愈者和死亡者）在人群中的比例
- G：隔离者（与病源接触过，可追踪到，并将其隔离的部分）在人群中的比例
- W：未隔离者（与病源接触过，未能追踪到，没有将其隔离的部分）在人群中的比例

3 总体假设

- 在新冠疫情爆发初期，由于病毒潜伏期的存在，社会各界对于新冠病毒传播的速度和危害程度认识不够充分，未能在第一时间采取有效措施。当人们发现被传染人数的迅速增加，科学界、医学界更多地了解到病毒的特性——传染性强并且潜伏期长后，政府出台各种强有力政策以控制病毒的进一步传播。对于我国的新冠病毒疫情，结合以上事实，并假设国家传染病疾控防治中心提供的数据真实可信，我们将新冠病毒的传播规律可以分为三个阶段：
- 干预前：接近于自然传播的传播模式。
- 过渡期：在公众开始意识到病毒的严重性，到政府采取有效措施前的一段时间内。
- 干预后：在介入人为因素之后的传播模式。

4 干预前模型

4.1 本地参数

- λ_1：每个病人平均每天有效接触（使人感染）的人数
- q：退出率，新冠病毒患者的日死亡率和日治愈之和
- L：（流入）流出人口占本地总人口的比例
- ε_1：处于潜伏期的病人的日发病率
- P：流入人口中带菌者所占的比例

4.2 本地假设

干预前（包括干预力度不大的阶段）的传播模型的相关假设：

- 将新冠病毒所有可能的传播途径都视为与病源的直接接触。
- 在疾病传播期内所考察的地区的总人数N视为常数，即认为本地区流入的人口与流出的人口数相等，时间一天为计量单位。
- 根据国家卫生部资料可知处于潜伏期的新冠肺炎病人与确诊新冠肺炎病人相比具有较弱的传染性，在本模型中视为不具有传染性。
- 假设潜伏期为 2-14 天，取其中位数（7 天）。

- 根据目前的医学调查资料，极少数康复者有复发情况，而根据资料显示，康复者康复之后，更注重自己的个人卫生并主动隔离自身，因此我们在本模型中假设康复者二度感染新冠病毒和将新冠病毒传染给其他人的概率为 0，他们已经退出传染体系，因此将他们归为“退出者”这一人群。
- 在新冠病毒传播期间，不考虑该时间段内的人口出生率和死亡率。对于由新冠病毒引起的死亡人数，也将其归为“退出者”这一人群。
- 将人群分为四类：S健康者、E潜伏期者、I病人、R退出者。
- 在病毒爆发初期，缺乏人工干预，新冠病毒传播接近于自然传播。

4.3 模型建立

在疫情爆发初期，存在统计上的误差等问题，并且由于病毒的传染性和潜伏周期等关键数据的缺失，使得获得的数据不够准确。这造成了疫情初期所建立的模型将与实际情况存在较大出入，对于病毒未来的预测帮助不大。因此这里仅对控前模型进行简单分析。

根据对上述符号的分析假设，可以对病毒传播的过程变化建立如下方程：

$$\frac{dS}{dt} = -\lambda_1 IS$$

$$\frac{dE}{dt} = \lambda_1 IS - \varepsilon_1 + LP - LE$$

$$\frac{dI}{dt} = \varepsilon_1 E - qI$$

$$\frac{dR}{dt} = qI$$

4.4 本地参数的确定

- λ_1：根据医学资料和有关数据推导而得。
- q：由该地区的医疗水平所决定。
- L：考虑地区的出入人口流动情况（主要由经济发达程度和交通状况决定），查有关资料得到。
- ε_1：根据医学研究和调查该地区的疫情发展状况可得。
- P：由流入该地区人群的地区分布情况和各其他地区的疫情决定。

5 干预后模型

5.1 本地参数

- λ_2：不可控人群（在后面的分析中可得到）在发病后到被隔离前平均每天接触的人和数目
- q：退出率，新冠病毒患者的日死亡率和日治愈之和
- L：（流入）流出人口占本地总人口的比例
- β：接触病源的人的发病率
- ε：每天由可控人群和不可控人群转化为病人的日转化率

5.2 本地假设

干预后的传播模型的相关假设：

- 由于新冠病毒的治愈疗程迄今还没有确切的资料，而且每一天都有一批人被治愈，故我们在计算退出率时并没有考虑疗程的问题。
- 被隔离人群完全断绝与外界的接触，不再具有传染性。
- 不考虑无症状感染者，即只要感染上新冠病毒的患者最后的都会表现出症状。
- 人群被划分为五类：S健康者，I病人，R退出者，G隔离者，W未隔离者。

5.3 模型建立

$$\frac{dS}{dt} = -\lambda_2 W\beta S$$

$$\frac{dI}{dt} = \varepsilon\beta(G+W) - qI$$

$$\frac{dR}{dt} = qI$$

$$\frac{dG}{dt} = \lambda_2\beta WS\frac{G}{G+W} - \beta\varepsilon G$$

$$\frac{dW}{dt} = \lambda_2\beta\varepsilon S\frac{W}{G+\varepsilon} - \beta\varepsilon W$$

在干预前模型中，我们对模型和数据进行了进一步的分析，发现这个模型存在以下问题：

- 该模型中，没有充分考虑疑似病例，即“疑似者”和“隔离者”的之间的关系不明确。
- 从收集到的数据中我们无法得到有关隔离者和未被隔离者的信息，因此无法对其做出分析。

以下，我们将对以上问题进行改进。

5.4 本地参数的确定

- λ_2：根据查阅相关医学资料而得。
- q：由该地区的医疗水平所决定。
- β：根据医学资料和有关数据推导而得。
- ε_1：根据医学研究和调查该地区疫情发展状况得。
- L：在干预后模型中，人口流动受到了一定的限制，根据不同地区的政策和文化性质可得该参数。

6 优化模型

在干预后模型的基础上，以下优化模型将修正先前提到的未考虑因素和部分不确定因素。明确了疑似者所指的范围，并将感染者进一步细分为轻度、重症和严重三个阶段。细化三类人群的传播感染率、治愈率，分别设定参数，使模型更加接近实际情况。

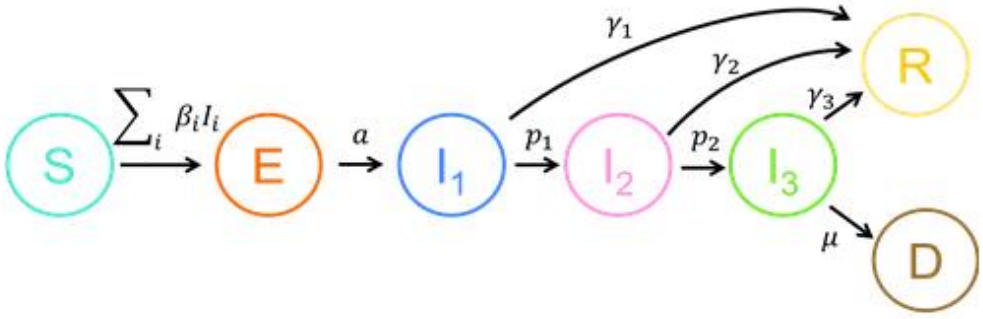

图 1：个体转化情况示意图

6.1 本地定义

- S：易感人群
- E：感染人群-感染但尚未感染或有症状
- I_i：严重等级为i的受感染个体。 严重程度随着i的增加而增加，并且我们假设在$i+1$层的个体由i层的个体发展而来
- I_1：轻度感染者
- I_2：严重感染者
- I_3：重症感染者
- R：已经治愈因此不再被感染的个体
- D：死亡个体
- $N = S + E + I_1 + I_2 + I_3 + R + D$：总人口数

6.2 本地参数

- β_i：在i 阶段接触易感人群并且感染的比率
- a：暴露人群的感染比率
- γ_i：在i阶段感染个体的治愈率
- p_i 第l_i：阶段感染者发展为第l_{i+1}阶段的比率
- μ：重症患者的死亡率

6.3 本地假设

- 该模型被公式化为一个微分方程系统，因此输出表示每个量的期望值。 它没有考虑到随机事件，因此即使流行病降至非常低的水平，流行也不会灭绝（除非停止干预，在这种情况下，每个州的人数都将四舍五入到最接近的整数）。
- 后一阶段的患者一定由前一阶段的患者发展而来。
- 只有处于严重阶段的人死亡。
- 各阶段的患者均具有相同的传播率。
- 易感人群拥有相同易感性。
- 短时间内包括死亡个体在内的总人口数保持不变。

6.4 模型建立

$$\frac{dS}{dt} = -(\beta_1 I_1 - \beta_2 I_2 - \beta_3 I_3)$$

$$\frac{dE}{dt} = (\beta_1 I_1 + \beta_2 I_2 + \beta_3 I_3)S - aE$$

$$\frac{dI_1}{dt} = aE - (\gamma_1 + p_1)I_1$$

$$\frac{dI_2}{dt} = p_1 I_1 - (\gamma_2 + p_2)I_2$$

$$\frac{dI_3}{dt} = p_2 I_2 - (\gamma_3 + \mu)I_3$$

$$\frac{dR}{dt} = \gamma_1 I_1 + \gamma_2 I_2 + \gamma_3 I_3$$

$$\frac{dD}{dt} = \mu I_3$$

6.5 本地参数的确定

由于实际获得的数据不能直接带入模型，我们设以下中间参数进行转化。

- $IncubPeriod$：平均潜伏期天数
- $DurMildInf$：轻度感染的平均持续天数
- $FracMild$：轻度感染者在人群中的比例
- $FracSevere$：严重感染者在人群中的比例
- $FracCritical$：重症感染者在人群中的比例
- CFR：病死率
- $DurHosp$：重症感染者的平均住院天数
- $TimeICUDeath$：ICU 患者的住院天数（直到死亡或康复）

$$a = \frac{1}{IncubPeriod}$$

$$\gamma_1 = \frac{1}{DurMildInf} \times FracMild$$

$$p_1 = \frac{1}{DurMildInf} - \gamma_1$$

$$p_2 = \frac{1}{DurHosp} \times \frac{FracCritical}{FracSevere + FracCritical}$$

$$\gamma_2 = \frac{1}{DurHosp} - p_2$$

$$u = \frac{1}{TimeICUDeath} \times \frac{CFR}{FracCritical}$$

$$\gamma_3 = \frac{1}{TimeICUDeath} - u$$

7 干预效果模拟分析

假设干预措施对处在I_1阶段的轻症患者有效，能降低轻症患者的人传人的比率β_1。结合实际情况，即较危重的患者送往医院等处隔离，出于简化模型的考虑，不考虑干预对处在I_2、I_3阶段的患者的效果。

考察开始干预的时间以及干预的有效程度，即能降低传染比例的多少两个参数对疫情防治的影响。

模拟结果如下：

其中深红线为不加干预条件的自然条件下，现存患病总人数（$I_1 + I_2 + I_3$）随时间的变化；橙色线为加干预条件后的患病总人数随时间的变化。

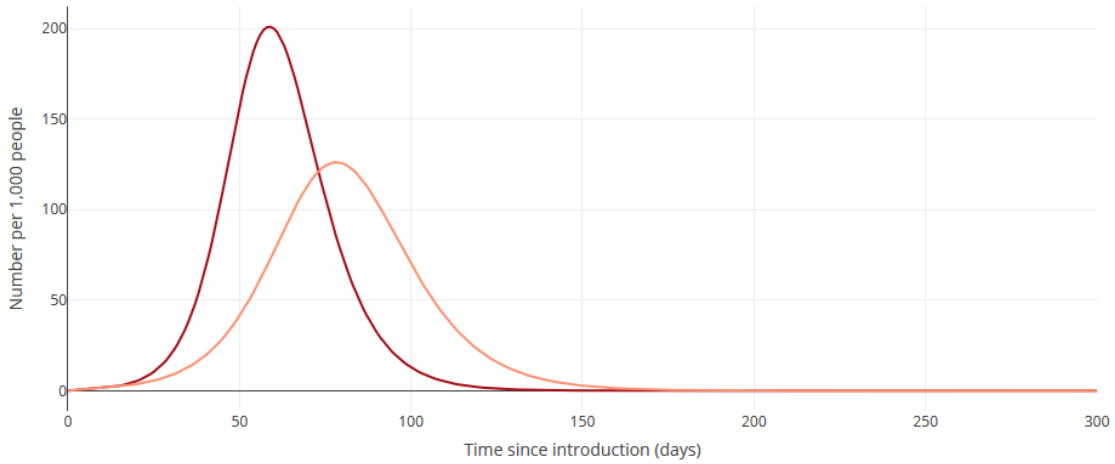

图 2：10 天后开始干预，降低 30%的传染比例模拟结果图

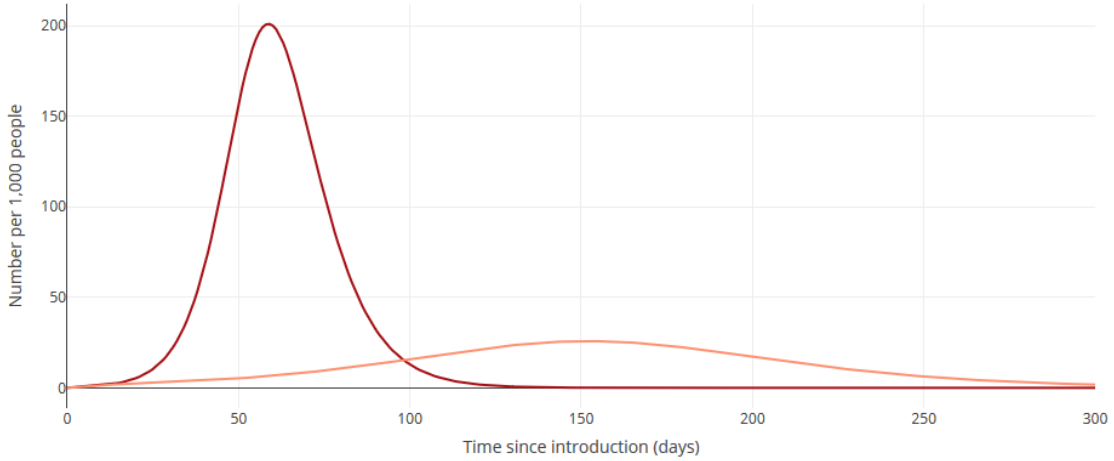

图 3：10 天后开始干预，降低 60%的传染比例模拟结果图

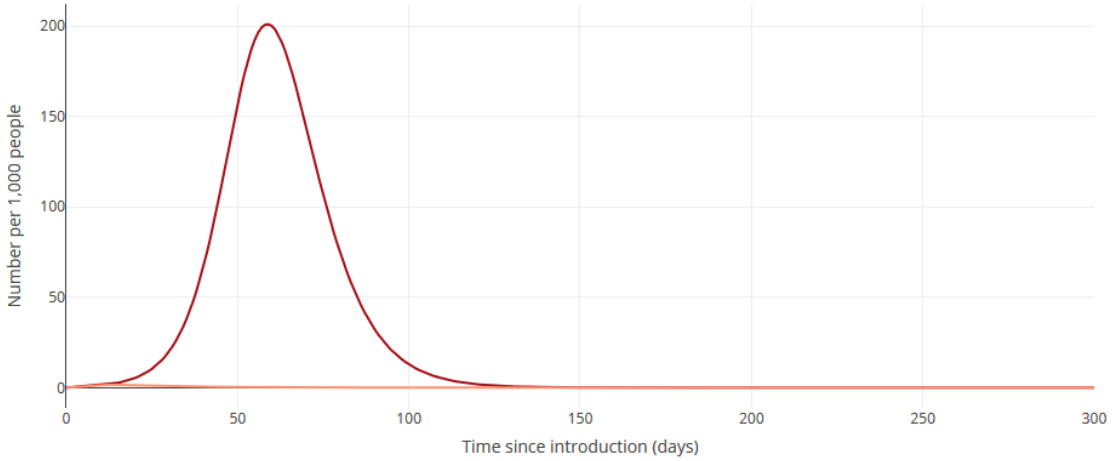

图 4：10 天后开始干预，降低 90%的传染比例模拟结果图

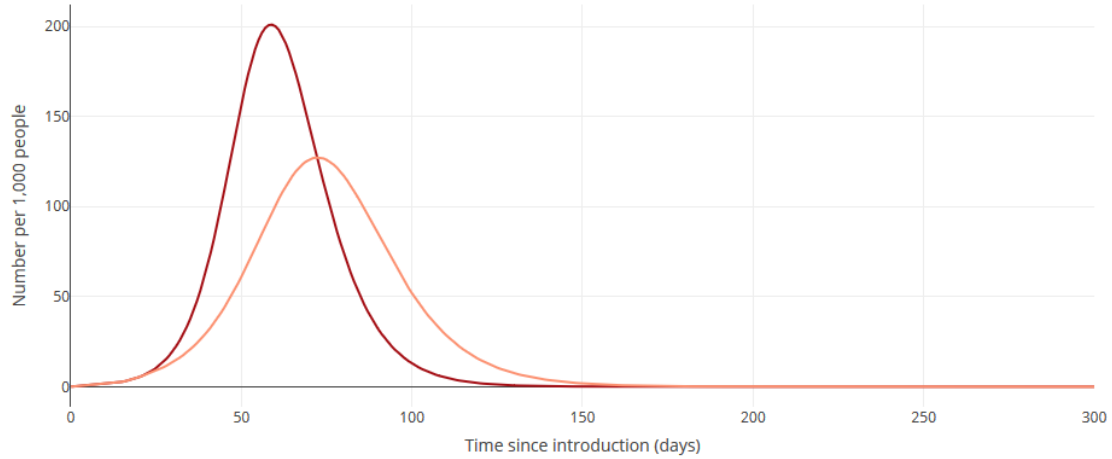

图 5：20 天后开始干预，降低 30%的传染比例模拟结果图

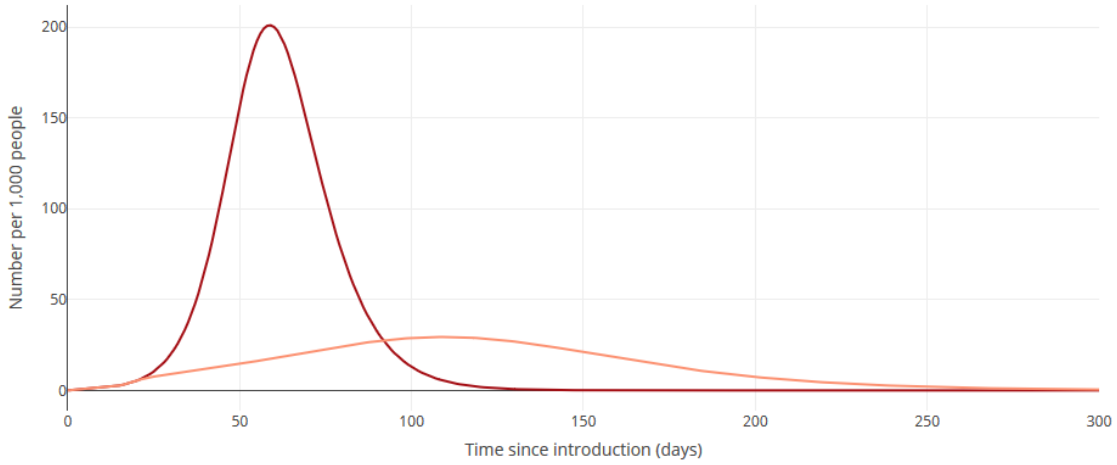

图 6：20 天后开始干预，降低 60%的传染比例模拟结果图

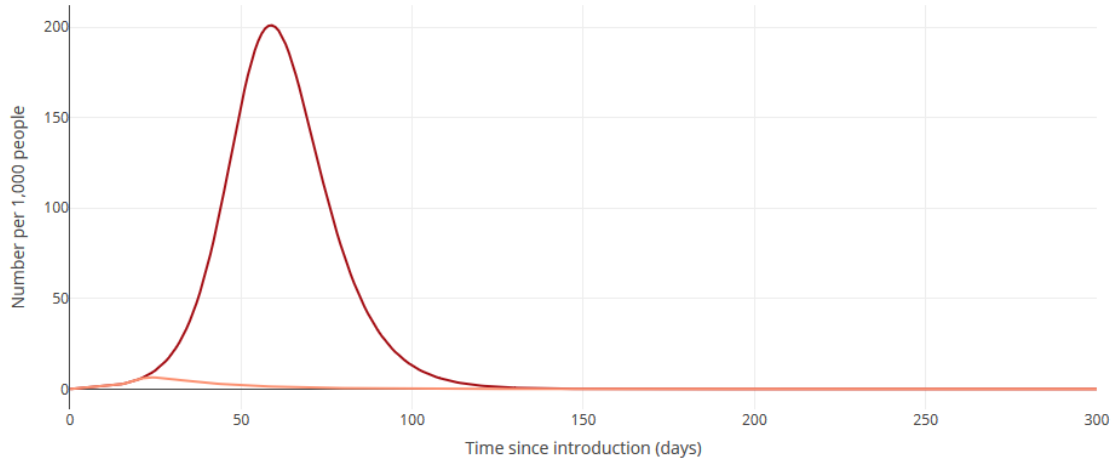

图 7：20 天后开始干预，降低 90%的传染比例模拟结果图

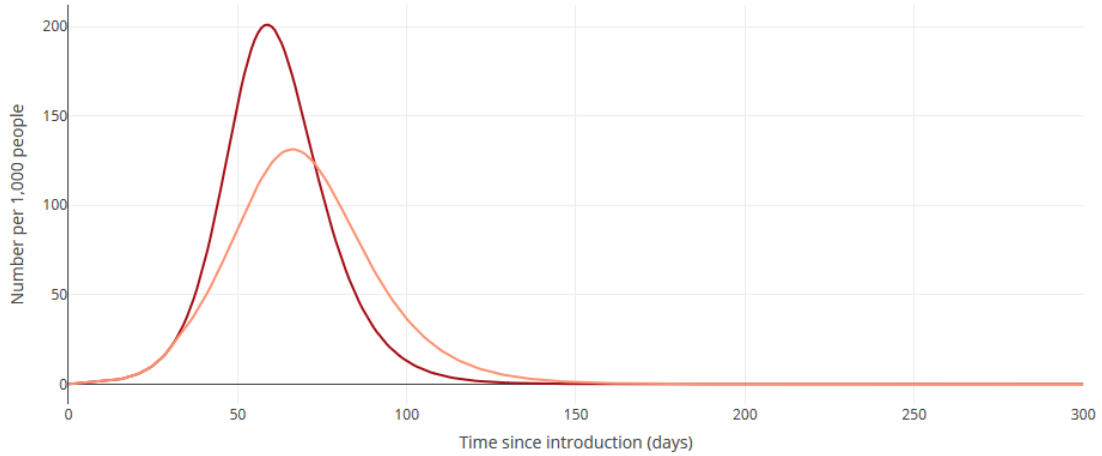

图 8：30 天后开始干预，降低 30%的传染比例模拟结果图

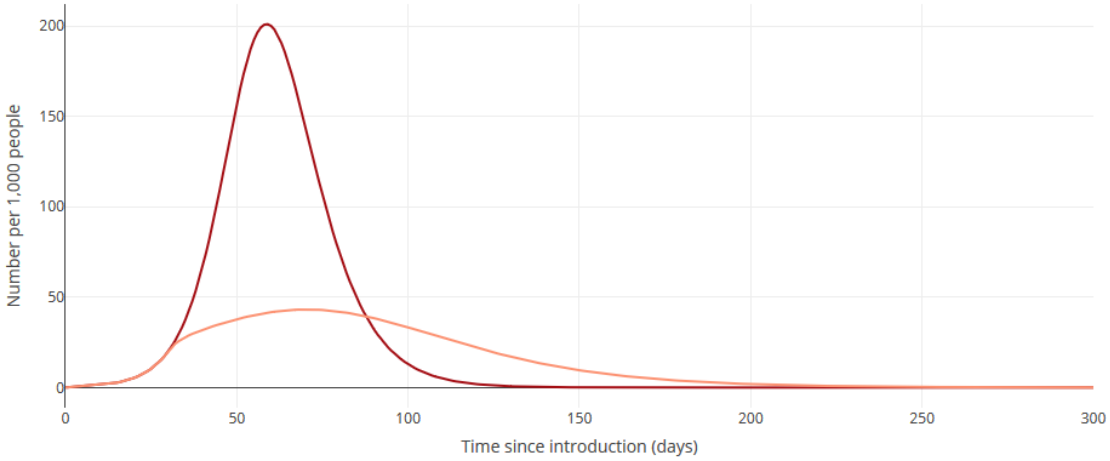

图 9：30 天后开始干预，降低 60%的传染比例模拟结果图

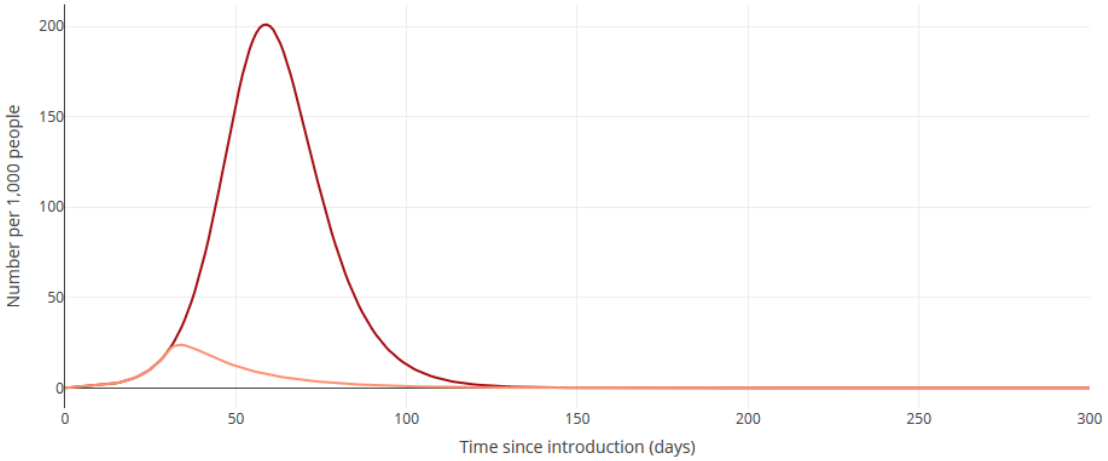

图 10：30 天后开始干预，降低 90%的传染比例模拟结果图

由以上模拟结果，可以直观地得出尽早出台政策干预、控制疫情以及加大疫情管控力度对于控制疫情所作出的贡献。

一、横向对比。同时间开始干预的情况下，当传染比率降低 30%时，现存患者数的峰值并未降低很多，而当传染比率降低 60%后，已能大幅降低现存患者数的峰值至超过 5 倍。这意味着，对于仅有有限的医疗资源，有限的承载病人数的医院而言，可大幅减轻其收治病患的压力。可见，积极有效的干预措施，对于疫情防控的效果是非常显著的。而群体免疫等消极的防控措施是不可取的。

二、纵向对比。 相同干预强度的情况下（这里取 60%），当第 30 天采取干预时，尽管采取了较为有力的措施来减少传染比率，但是疫情持续天数和传播人数的峰值依旧不是很乐观，而当干预开始的天数为 20 天时，感染人数的峰值小幅度下降了约 25%，当干预开始的天数为 10 天时，感染人数的峰值较 30 天开始干预的情况减少了将近 1 倍。可见，尽早采取管控措施对于疫情的防控是至关重要的。

综合以上分析，可知，早发现、早隔离以及强有力的干预措施均可非常显著地改善疫情防控的效果。给我们的启示是，对一个未知的传染病，早期便需要非常重视，尽早地干预。居家隔离、出门戴口罩等均能为疫情的控制作出贡献。

8 预测结果及分析

8.1 湖北、武汉

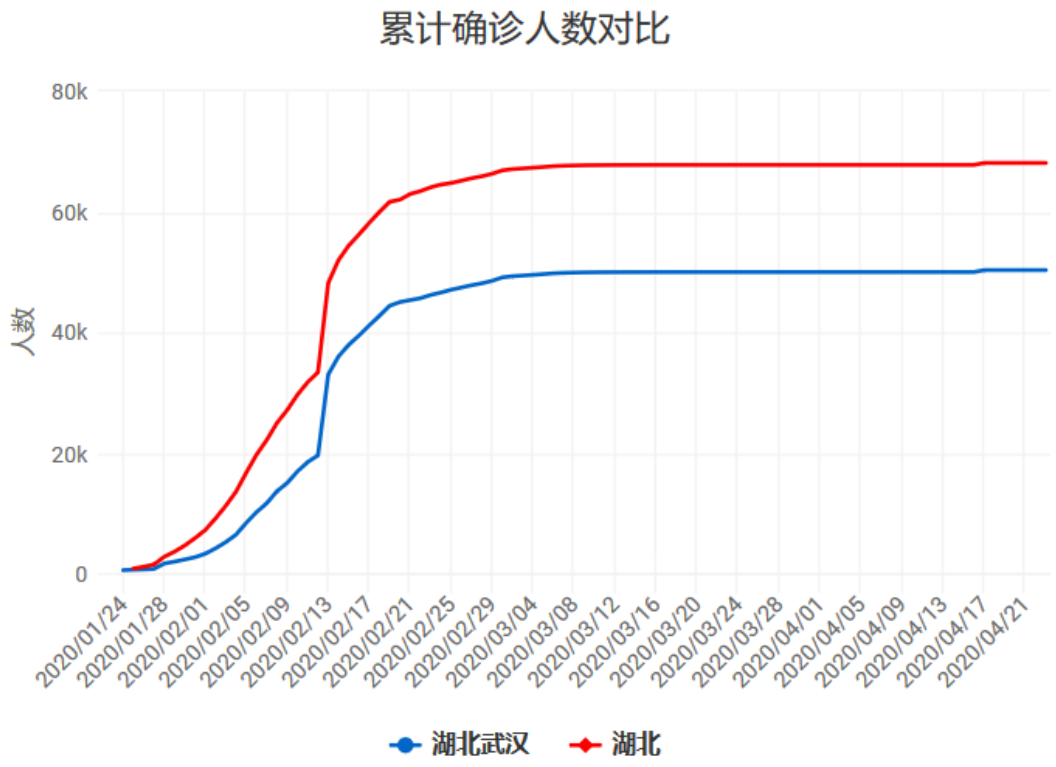

图 11： 湖北、武汉累计确诊人数图

由图可知，湖北省的新增确诊人数在 3 月 8 日前呈上升趋势，且增长速度在 2 月 13 日达到峰值，之后增长速度逐渐减小，说明疫情工作开始见效；在 3 月 8 日后，累计确诊人数基本不变，说明基本上没有新增病例，疫情得到了有效的控制。由于武汉市确诊人数占湖北省的大多数，武汉市新增确诊人数与湖北省变化趋势基本一致，其增长速度在 2 月 13 日前后达到峰值，之后一直下降至零。

8.2 美国

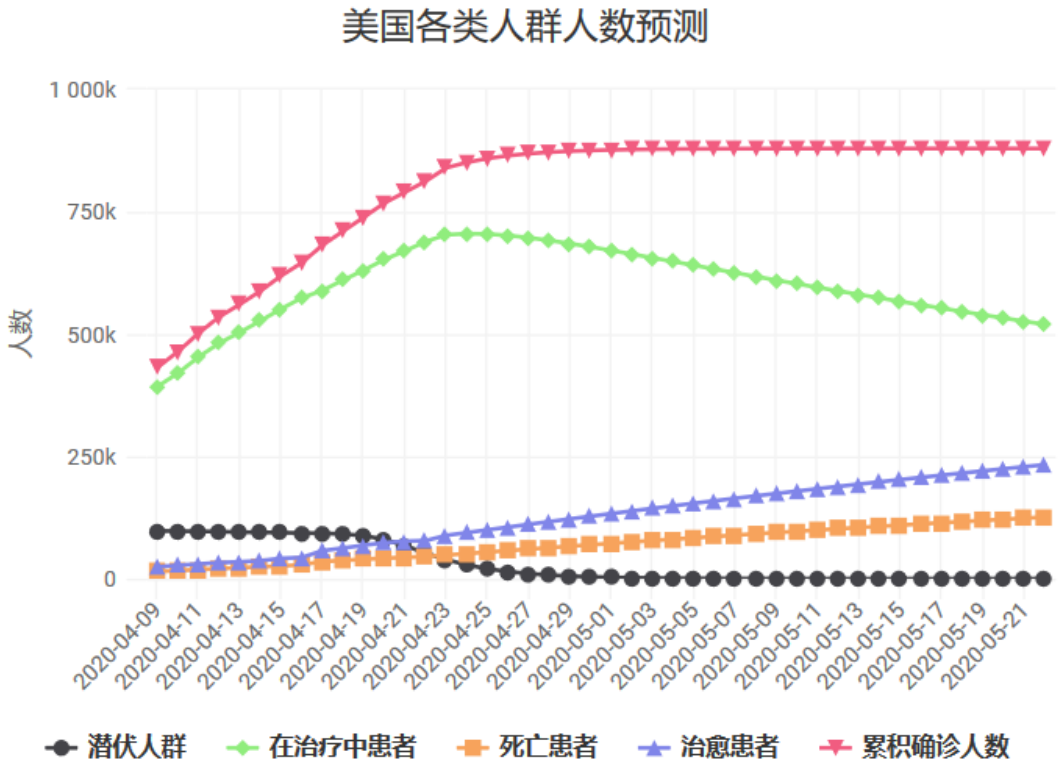

图 12： 美国各类人群人数预测图

根据我们已经建立的模型和现阶段美国已有的数据对疫情进行预测，生成上图预测结果，可以看到美国在 4 月 24 日，迎来了累计确诊人数的拐点，可判断出疫情得到了初步控制，每日新增确诊人数降低，之后每日新增确诊人数一路下降，到 5 月 9 日可以看到的是每日确诊人数几乎不再增加，而在治疗中的患者总人数自 4 月 24 日开始下降，进一步说明疫情得到了控制。

8.3 英国

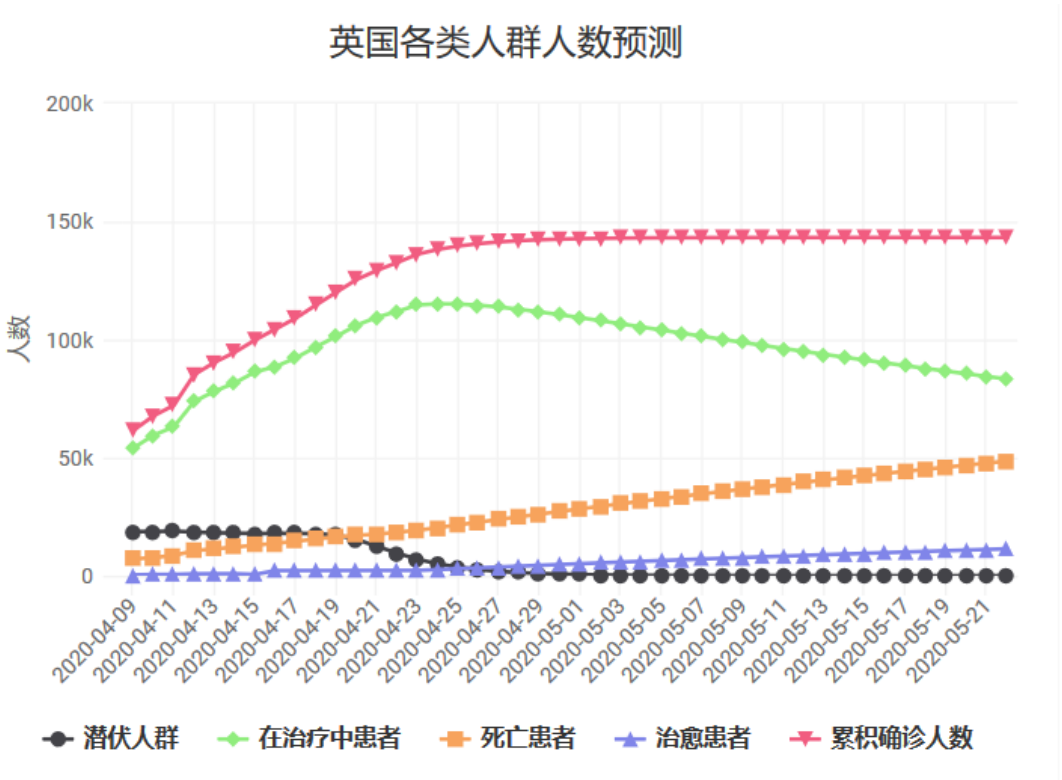

图 13：英国各类人群人数预测图

根据我们已经建立的模型和英国已经有的数据进行预测。预测结果曲线同美国的曲线非常相似，预测得在 4 月 23 日时迎来拐点。此后，现存患者将逐步减少，医疗资源的挤兑现象将有所缓解。仅从当前模型的预测结果得出结论，预期大致在 5 月中旬开始，英国将能较大程度地控制住疫情。

8.4 日本

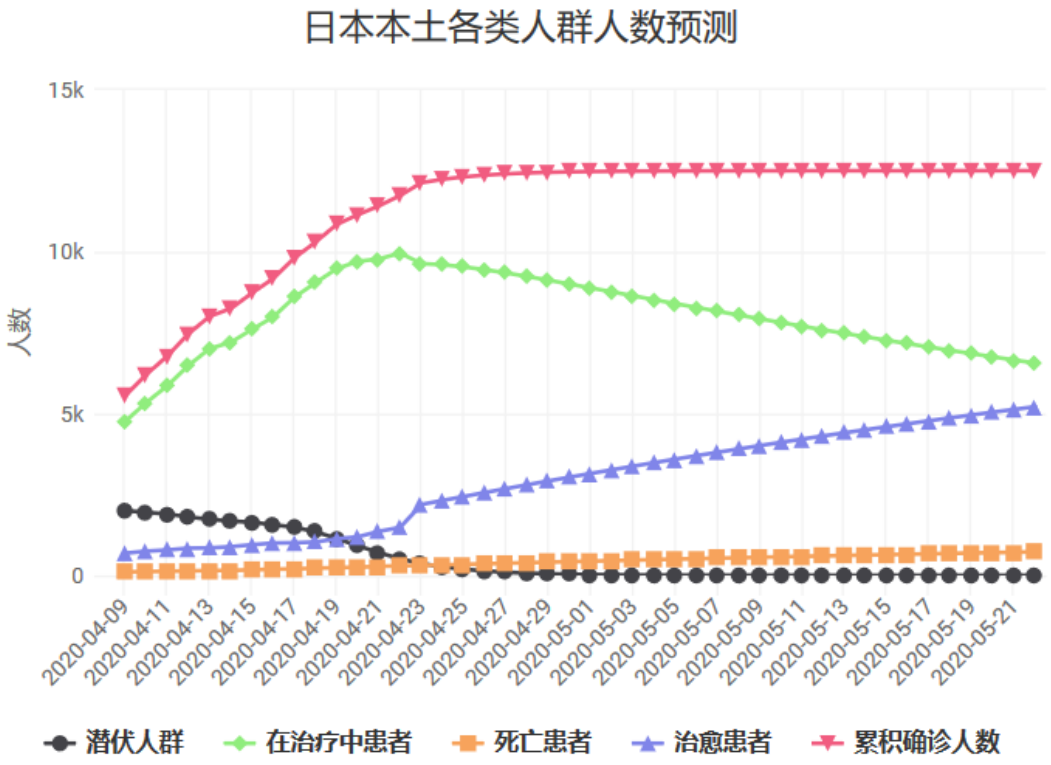

图 14：日本各类人群人数预测图

根据我们的模型预测，日本累计确诊人数在短时间内将持续上升，但增长速度较为平缓，预计到 5 月 1 日前后将得到有效控制，新增病例将逐渐较少到零。治愈患者和死亡患者会平稳上升，由于潜伏人群的减小，收治患者将越来越少，在 4 月 21 日达到峰值后出现拐点，之后预计将逐渐减小。

8.5 累积和新增境外输入病例趋势

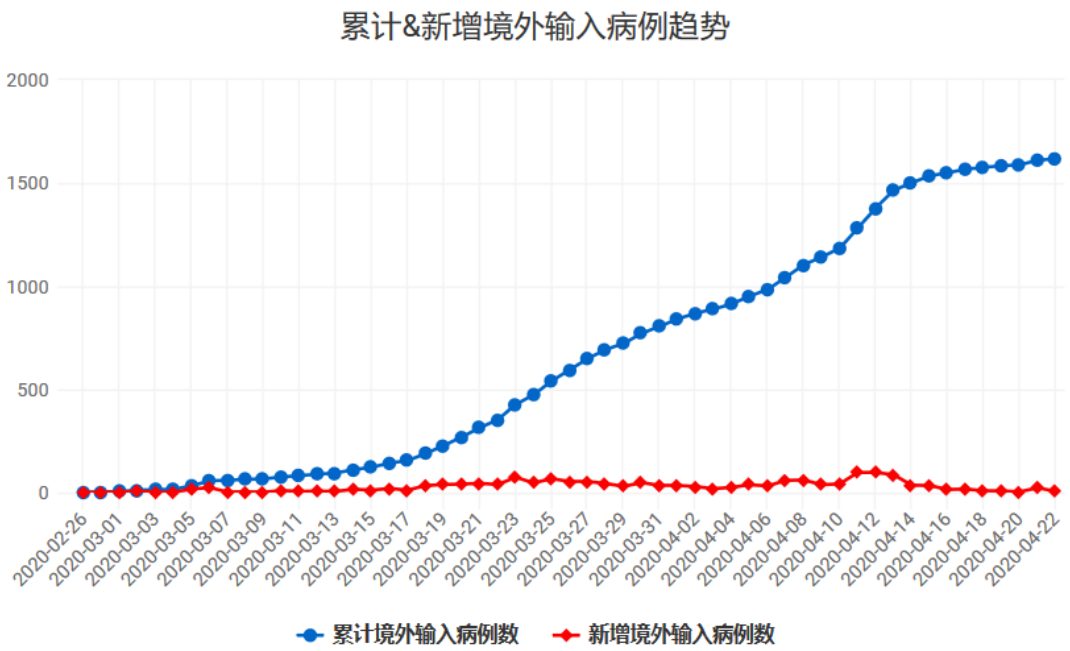

图 15：我国累计和新增境外输入病例趋势图

结合目前我国的疫情形势，新增病例主要来源于境外输入。且随着俄罗斯等地开始爆发疫情，境外输入的病例可能将在未来一段时间内继续持续上升。我国的疫情防治重心已从控制本土新增转向严防境外输入，未来对境外输入的严防严控不可掉以轻心，在全球疫情不容乐观的当下，对新冠病毒的战役仍在进行当中。

9 模型评价

9.1 优点

在本文中我们采用了经典的传染病模型，即 SEIR 模型。SEIR 模型描述了人群之中人口的迁移，感染人群在总人群中的比率对感染率的影响，是比较成熟的传染病模型。

在原始模型的基础上，我们还增添了三个特殊的临床阶段，意图来说明每一个阶段人群的康复率、传染率和死亡率是不同的这一临床现象。

9.2 不足

- 模型没有考虑无症状感染者的因素。在易感人群S变为感染人群E之后，一部分的人会成为无症状感染者。对于无症状感染者的考量需要在模型中进一步补充。
- 模型中没有考虑温度对因素对病毒活性的影响。
- 对于不同地区的人口具有不同的流动性，干预前模型没有考虑人口流动因素的影响。
- 模型中没有考虑地区医疗水平、医疗资源的影响。

9.3 未来工作

- 对不足中提到的欠考虑的因素加以考量，试图增加参数、修改参数来刻画以上影响因素。
- 在干预效果的模拟分析中，细化、多样化参数的选择，提高采样密度，以得到更为精确的干预下病毒发展规律。
- 对疫情发展的趋势、各种参数的影响做可视化分析工具。

参考文献与数据来源

1. 姜启源, 谢金星, 叶俊 (2011) 数学模型. 第四版, 高等教育出版社, 北京.
2. G Chowell, C Castillo-Chavez, PW Fenimore, CM Kribs-Zaleta, L Arriola, JM HymanModel parameters and outbreak control for SARS Emerg Infect Dis, 10 (2004), pp. 1258-1263
3. O Diekmann, JAP Heesterbeek Mathematical epidemiology of infectious diseases: model building, analysis and interpretation, John Wiley, Chichester (2000)
4. https://github.com/GuangchuangYu/nCov2019
5. https://github.com/canghailan/Wuhan-2019-nCoV
6. https://news.qq.com/zt2020/page/feiyan.htm#/global

附录

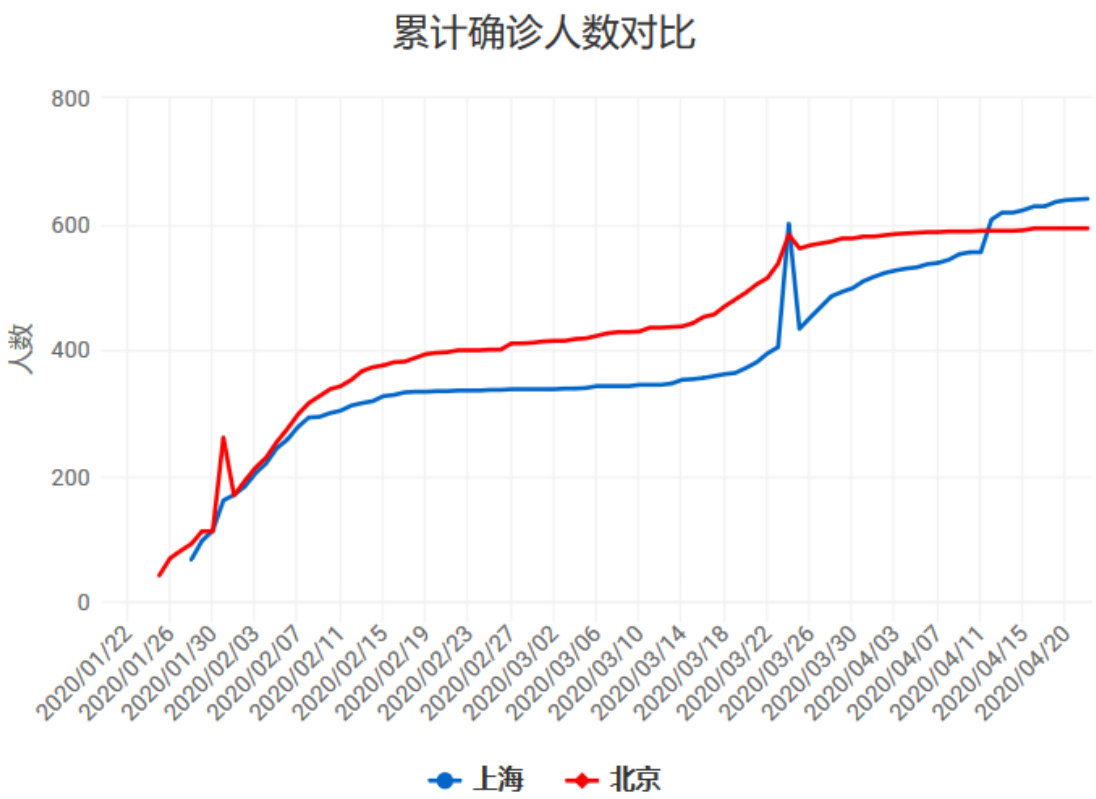

图 16：上海、北京累计确诊人数对比图

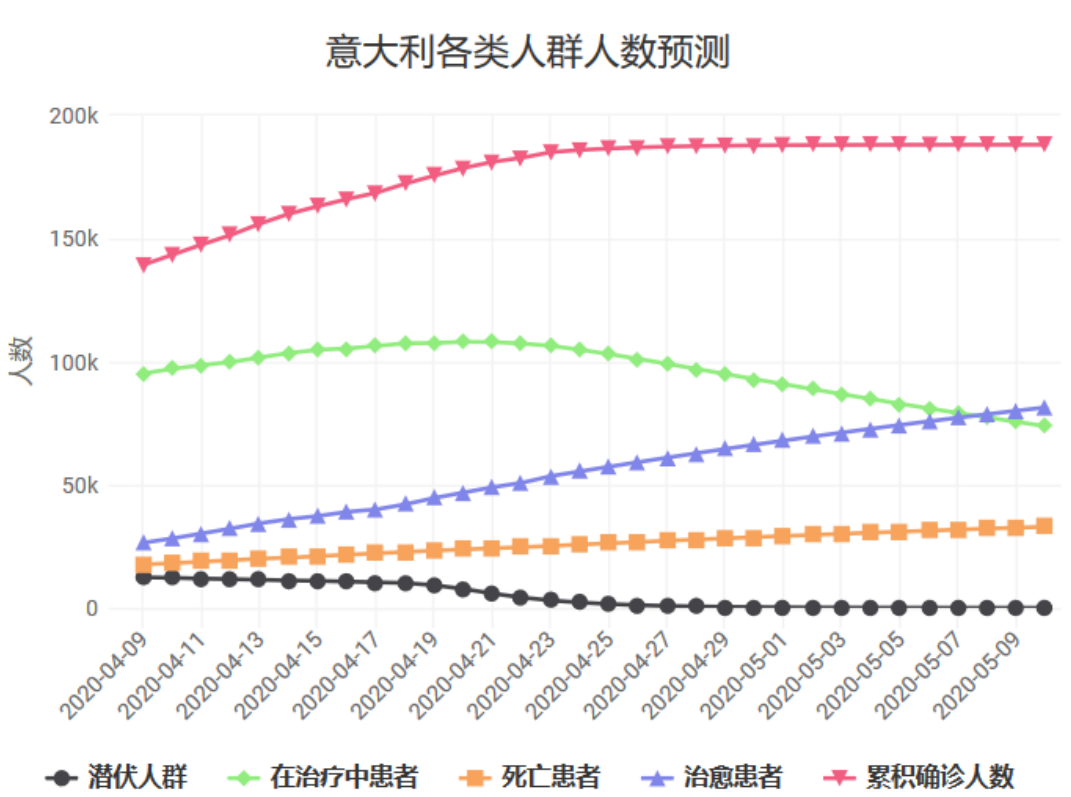

图 17：意大利各类人群人数预测图

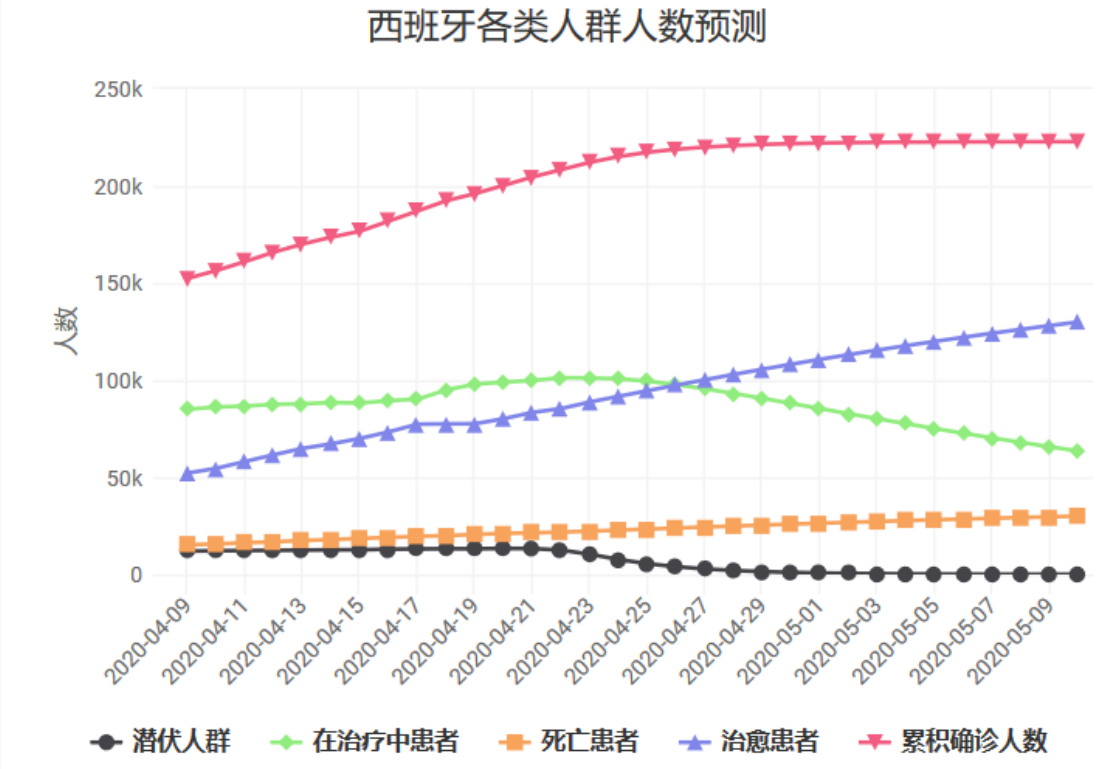

图 18：西班牙各类人群人数预测图

模型求解 R 语言代码：

```
SetODEs_SEIR=function(t,y,p){
  S = y[1]
  E0 = y[2]
  E1 = y[3]
  I0 = y[4]
  I1 = y[5]
  I2 = y[6]
  I3 = y[7]
  R = y[8]
  D = y[9]

  with(as.list(p),{

    seas=(1 + seas.amp*cos(2*pi*(t-seas.phase)/365))

    dS.dt = -(be*E1+b0*I0+b1*I1+b2*I2+b3*I3)*S*seas
    dE0.dt=(be*E1+b0*I0+b1*I1+b2*I2+b3*I3)*S*seas-a0*E0
    dE1.dt=a0*E0-a1*E1
    dI0.dt=f*a1*E1-g0*I0
    dI1.dt=(1-f)*a1*E1-g1*I1-p1*I1
    #dI2.dt=p1*I1-g2*I2-p2*I2
    dI2.dt=p1*I1-g2*I2-progress_fun(I2,p2,p2max,hcap)
    #dI3.dt=p2*I2-g3*I3-u*I3
    dI3.dt=progress_fun(I2,p2,p2max,hcap)-g3*I3-deaths_fun(I3,u,umax,ic
ap)
    dR.dt=g0*I0+g1*I1+g2*I2+g3*I3
    dD.dt=deaths_fun(I3,u,umax,icap)

    return(list(c(dS.dt, dE0.dt, dE1.dt,dI0.dt,dI1.dt, dI2.dt, dI3.dt,
dR.dt, dD.dt)))
  })
}


progress_fun=function(I2,p2,p2max,hcap){
  #progress=p2*I2/(1+(I2/hcap)^m)+(p2*hcap+p2max*(I2-hcap))*(1-1/(1+(I2
/hcap)^m)) #continuous
  progress=ifelse(I2<hcap,p2*I2,p2*hcap+p2max*(I2-hcap)) #discrete
  return(progress)
}


deaths_fun=function(I3,u,umax,icap){
  #deaths=u*I3/(1+(I3/icap)^m)+(u*icap+umax*(I3-icap))*(1-1/(1+(I3/icap)
```

```
^m)) #continuous
  deaths=ifelse(I3<icap,u*I3,u*icap+umax*(I3-icap)) #discrete
  return(deaths)
}


GetSpread_SEIR = function(p,Tmax,y0){
  t = seq(from=0, to=Tmax, by=1)
  out = ode(y=y0, times=t, func=SetODEs_SEIR, parms=p)
  df = as.data.frame(out)
  return(df)
}


GetModelParams = function(input){

  IncubPeriod=input$IncubPeriod  #Incubation period, days
  DurMildInf=input$DurMildInf #Duration of mild infections, days
  FracSevere=input$FracSevere/100 #Fraction of infections that are seve
re
  FracCritical=input$FracCritical/100 #Fraction of infections that are
critical
  FracMild=1-FracSevere-FracCritical  #Fraction of infections that are
mild
  ProbDeath=input$ProbDeath  #Probability of dying given critical infec
tion
  CFR=ProbDeath*FracCritical/100 #Case fatality rate (fraction of infec
tions resulting in death)
  TimeICUDeath=input$TimeICUDeath #Time from ICU admission to death, da
ys
  DurHosp=input$DurHosp #Duration of hospitalization, days

  FracProgressNoCare = input$FracProgressNoCare/100 # % of symptomatic
infections that would progress to critical without any hospital care
  FracDieNoCare=input$FracDieNoCare/100 # % of critical infections that
 would die without ICU care (can't be exactly 100%)


  capParams=SetHospCapacityOverflow(input)
  icap=unname(capParams["icap"])
  hcap=unname(capParams["hcap"])

  N=input$N



  pClin=c(IncubPeriod=IncubPeriod, DurMildInf=DurMildInf,FracMild=FracM
ild, FracSevere=FracSevere,FracCritical=FracCritical,CFR=CFR,TimeICUDea
```

```
th=TimeICUDeath,DurHosp=DurHosp,FracAsym=FracAsym,PresymPeriod=PresymPe
riod,DurAsym=DurAsym,FracProgressNoCare=FracProgressNoCare,FracDieNoCar
e=FracDieNoCare)

  # Turn these clinical parameters into the rate constants of the model

  pModel=GetParams_SEIR(pClin)

  pModel=c(be=be,b0=b0,b1=b1,b2=b2,b3=b3,pModel)

  pModel=c(pModel,seas.amp=seas.amp, seas.phase=seas.phase, icap=icap,
hcap=hcap)


  return(list("N"=N,"pModel"=pModel))
}



GetParams_SEIR = function(pClin){

  with(as.list(pClin),{

    a1=min(10^6,1/PresymPeriod) #presymptomatic period of transmission
    a0=min(10^6,(IncubPeriod-PresymPeriod)^(-1)) # true latent period,
avoid infinity when no presymptomatic phase

    f=FracAsym

    g0=1/DurAsym

    g1=(1/DurMildInf)*FracMild
    p1=(1/DurMildInf)-g1

    p2=(1/DurHosp)*(FracCritical/(FracSevere+FracCritical))
    g2=(1/DurHosp)-p2

    p2max=FracProgressNoCare*g2/(1-FracProgressNoCare)


    if(FracCritical==0){
      u=0
    }else{
      u=(1/TimeICUDeath)*(CFR/FracCritical)
    }

    g3=(1/TimeICUDeath)-u
```

```
    if(FracDieNoCare==0){
      umax=0
    }else{
      umax=FracDieNoCare*g3/(1-FracDieNoCare)
    }


    return(c(a0=a0,a1=a1,f=f,g0=g0,g1=g1,g2=g2,g3=g3,p1=p1,p2=p2,u=u,p2
max=p2max,umax=umax))
  })

}



GetRo_SEIR = function(p,N){

  with(as.list(p),{

    Ro=N*((be/a1)+f*(b0/g0)+(1-f)*((b1/(p1+g1))+(p1/(p1+g1))*(b2/(p2+g2)
+ (p2/(p2+g2))*(b3/(u+g3)))))

    return(Ro)
  })

}


Getr_SEIR = function(p,N){


  with(as.list(p),{

    #  Compute the maximum eigenvalue, corresponding to r, and the corr
esponding eigenvector, which gives the ratios of the numbers of individ
uals in each class during the early growth phase
    # matrix representation of the linearized system when S=N
    JacobianMat=rbind(c(-a0, N*be, N*b0, N*b1, N*b2, N*b3, 0, 0),
                      c(a0, -a1, 0, 0, 0, 0, 0, 0),
                      c(0, a1*f, -g0, 0, 0, 0, 0, 0),
                      c(0, a1 - a1*f, 0, -p1-g1, 0, 0, 0, 0),
                      c(0, 0, 0, p1, -p2-g2, 0, 0, 0),
                      c(0, 0, 0, 0, p2, -u-g3, 0, 0),
                      c(0, 0, g0, g1, g2, g3 , 0, 0),
                      c(0, 0, 0, 0, 0, u, 0, 0)
    )

    eig=eigen(JacobianMat)
```

```
    eig$values=Re(eig$values) #sometimes it add zero complex parts
    r=max(eig$values)
    MaxEigenVector=eig$vectors[,which.max(eig$values)]
    MaxEigenVector=MaxEigenVector/MaxEigenVector[length(MaxEigenVector)]
 #normalize to deaths
    MaxEigenVector=Re(MaxEigenVector)
    DoublingTime=log(2)/r


    return(list("r"=r,"DoublingTime"=DoublingTime,"MaxEigenVector"=MaxE
igenVector))

  })

}


SetHospCapacity=function(input){

  AvailHospBeds=input$HospBedper*(100-input$HospBedOcc*(1+input$IncFluO
cc/100))/100 #Available hospital beds per 1000 ppl in US based on total
 beds and occupancy
  AvailICUBeds=input$ICUBedper*(100-input$ICUBedOcc*(1+input$IncFluOcc/
100))/100 #Available ICU beds per 1000 ppl in US, based on total beds a
nd occupancy. Only counts adult not neonatal/pediatric beds
  ConvVentCap=input$ConvMVCap #Estimated excess # of patients who could
 be ventilated in US (per 1000 ppl) using conventional protocols
  ContVentCap=input$ContMVCap #Estimated excess # of patients who could
 be ventilated in US (per 1000 ppl) using contingency protocols
  CrisisVentCap=input$CrisisMVCap #Estimated excess # of patients who c
ould be ventilated in US (per 1000 ppl) using crisis protocols

  capParams=c("AvailHospBeds"=AvailHospBeds,"AvailICUBeds"=AvailICUBeds,
"ConvVentCap"=ConvVentCap,"ContVentCap"=ContVentCap,"CrisisVentCap"=Cri
sisVentCap)

  return(capParams)
}


SimSEIR = function(input){

  ParamStruct=GetModelParams(input)
  pModel=ParamStruct$pModel
  N=ParamStruct$N
  Tmax=input$Tmax
```

```
  # Set initial conditions and time interval
  E00=input$InitInf
  S0 = N-E00
  y0 = c(S=S0, E0=E00,  E1=0, I0=0, I1=0, I2=0, I3=0, R=0, D=0)

  #get Ro and r values
  Ro=GetRo_SEIR(pModel,N)
  r.out=Getr_SEIR(pModel,N)
  r=r.out$r
  DoublingTime=r.out$DoublingTime

  #run ODEs
  out.df=GetSpread_SEIR(pModel,Tmax,y0)

  if(input$AllowSeason=="Yes"){

    Ro.Season=GetRo_SEIR_Season(pModel,N)

    return(list("out.df"=out.df,"N"=N,"Ro"=Ro,"Ro.Season"=Ro.Season,"r"
=r,"DoublingTime"=DoublingTime))

  }else{
    return(
      list("out.df"=out.df,"N"=N,"Ro"=Ro,"r"=r,"DoublingTime"=DoublingT
ime))

  }

}
```