

核酸检测中的混检方案

第二十四届华东杯大学生数学建模邀请赛

2022.5

摘要

在新冠疫情席卷全球背景下，为了减小核酸检测次数，加快核酸检测速度，进而及时有效地控制疫情，往往采用混合检测的方法。

本文首先建立单轮检测模型，考虑单个人的平均核酸检测次数及其期望，在已知或估计总人口数和感染病毒的人口数的基础上，求解混合检测每组的人数，使得核酸检测的总次数最小。接着将该模型应用于上海市和吉林省长春市的新冠疫情情况，绘制了核酸检测次数随着混合检测每组的人数的变化图。平衡了理论结果和现实操作后，选择每组 10 人的混合检测是合理的。

在单轮检测模型的基础上，本文又建立了多轮检测模型。在已知或估计总人口数、感染病毒的人口数、假阴性概率、假阳性概率、传染系数、痊愈率的基础上，使用基于单轮检测模型的算法计算混合检测每组的人数，并手动调整与搜索核酸检测的次数和间隔时间，使用算法模拟每个人的感染情况、隔离情况、复核情况，并统计是否感染病毒与是否被隔离四种状态的人数。接着将该模型应用于上海市的新冠疫情影响情况，

目录

1	问题的提出	3
2	问题的初步分析	3
3	单轮检测模型	4
3.1	模型的假设	4
3.2	符号的说明	4
3.3	模型的建立 ^[1]	4
3.4	模型的应用	5
4	多轮检测模型	6
4.1	问题的分析	6
4.2	模型的假设	6
4.3	符号的说明	6
4.4	模型的建立	7
4.4.1	初始状态	7
4.4.2	新增感染	7
4.4.3	混合检测	8
4.4.4	单人单管检测	8
4.4.5	痊愈	8
4.5	模型的应用	9
5	模型的评价与推广	10
5.1	模型的评价	10
5.1.1	优点	10
5.1.2	缺点	10
5.2	模型的推广	10
A	数据	12
A.1	上海市疫情数据	12
A.2	吉林省长春市疫情数据	13
B	代码	16
B.1	单轮检测模型	16
B.2	多轮检测模型	17
B.2.1	MATLAB	17
B.2.2	python	21

1 问题的提出

2019 年底、2020 年初开始，新冠疫情席卷全球。为了有效控制新冠病毒的传播，核酸检测是一个非常重要而有效的方法。但是由于核酸检测需要一定的时间，在进行大规模检测时，往往会因为来不及检测而耽误控制病毒传播的宝贵时间。

为此，通常采用混合检测的方法。将若干人组成一组，把他们的采样样本混合起来进行检测。若一组核酸检测的结果为阴性，那么这组样本对应的被检测者都是阴性的。若一组核酸检测的结果为阳性，那么这组样本对应的被检测者中至少有一人是阳性的，需要对这组样本的所有被检测者重新进行单人单管检测。

本文旨在回答以下几个问题：

1. 单轮检测中，理想的混检方案；
2. 根据现实数据，分析问题 1.
3. 多轮检测中，混检方案相对于单轮检测的调整。

2 问题的初步分析

本文将由浅入深地建立两个模型：单轮检测模型和多轮检测模型。

定义一轮检测包括如下两个步骤：先将若干人组成一组，进行混合检测；若一组核酸检测的结果为阳性，再对这组样本的所有被检测者进行单人单管检测。

在单轮检测模型中，仅进行一轮检测，不考虑在核酸检测结果发布前的新增病例以及核酸检测本身的假阳性和假阴性情况。对于给定的总人口数和初始感染人数，确定混合检测每组的人数，使得核酸检测的总次数最小。

在多轮检测模型中，考虑进行多轮检测，且考虑在核酸检测结果发布前的新增病例以及核酸检测本身的假阴性情况。对于给定的总人口数、初始感染人数、传染系数、假阴性概率，确定混合检测每组的人数、检测的次数、检测的间隔时间，使得核酸检测的总次数尽可能小、人群中剩余病毒携带者的概率尽可能低。

3 单轮检测模型

3.1 模型的假设

1. 仅进行一轮检测；
2. 病毒感染者等可能地出现在各组中；
3. 总人口数为常数；
4. 不考虑在核酸检测结果发布前的新增病例，即感染病毒的人口数为常数；
5. 不考虑核酸检测本身的假阳性和假阴性情况；
6. 核酸检测技术可以满足任意大的混合检测每组的人数。

3.2 符号的说明

符号	含义
n	总人口数
m	感染病毒的人口数
k	混合检测每组的人数

表 1: 单轮检测模型的符号说明

3.3 模型的建立 ^[1]

记随机变量 X 表示单个人的平均核酸检测次数。由于假设总人口数为常数，最小化核酸检测的总次数即最小化单个人的平均核酸检测次数，即最小化 $E[X]$ 。

又由于假设病毒携带者等可能地出现在各组中， n 为总人口数， m 为感染病毒的人口数，可知任意一个人为病毒感染者的概率为 $\frac{m}{n}$ ，任意一组核酸检测为阴性的概率为 $(1 - \frac{m}{n})^k$ 。进而得到 X 的分布：

$$P(X = \frac{1}{k}) = (1 - \frac{m}{n})^k$$
$$P(X = \frac{1}{k} + 1) = 1 - (1 - \frac{m}{n})^k$$

进而得到 X 的期望：

$$E[X] = 1 + \frac{1}{k} - (1 - \frac{m}{n})^k$$

需要最小化上式。特别地，当 $m \ll n$ 时，有

$$E[X] \approx 1 + \frac{1}{k} - (1 - k\frac{m}{n}) = \frac{1}{k} + k\frac{m}{n}$$

当 $k = \sqrt{\frac{n}{m}}$ 时，取得最小值 $2\sqrt{\frac{m}{n}}$ ，此时总核酸检测次数为 $2\sqrt{mn}$ ，且感染病毒的人口比例越高，单个人的核酸检测次数越高，符合预期。

3.4 模型的应用

本节将上述模型应用于上海市和吉林省长春市的新冠疫情情况。

根据上海市健康卫生委员会的疫情通报^[2]，最近一轮的上海疫情始于 2022 年 3 月 1 日，每日新增一万左右例感染者。再加上近期上海约两天进行一次核酸检测^[3]，可设置感染病毒的人口数为两万。

根据长春市健康卫生委员会的疫情通报^[4]，最近一轮的长春疫情始于 2022 年 3 月 5 日，每日新增一千左右例感染者。再加上近期长春约两天进行一次核酸检测^[45]，可设置感染病毒的人口数为两千。

进而得到参数设置与实验结果如下：

城市	n	m	m/n	最优 k	最优 $E[X]$	实际 k	实际 $E[X]$
上海	2.4×10^7 ^[6]	2×10^4	8.3×10^{-4}	35	0.0573	10 ^[7]	0.1083
长春	0.9×10^7 ^[8]	2×10^3	2.2×10^{-4}	68	0.0297	10 ^[9]	0.1022

表 2: 单轮检测模型应用的参数设置和实验结果

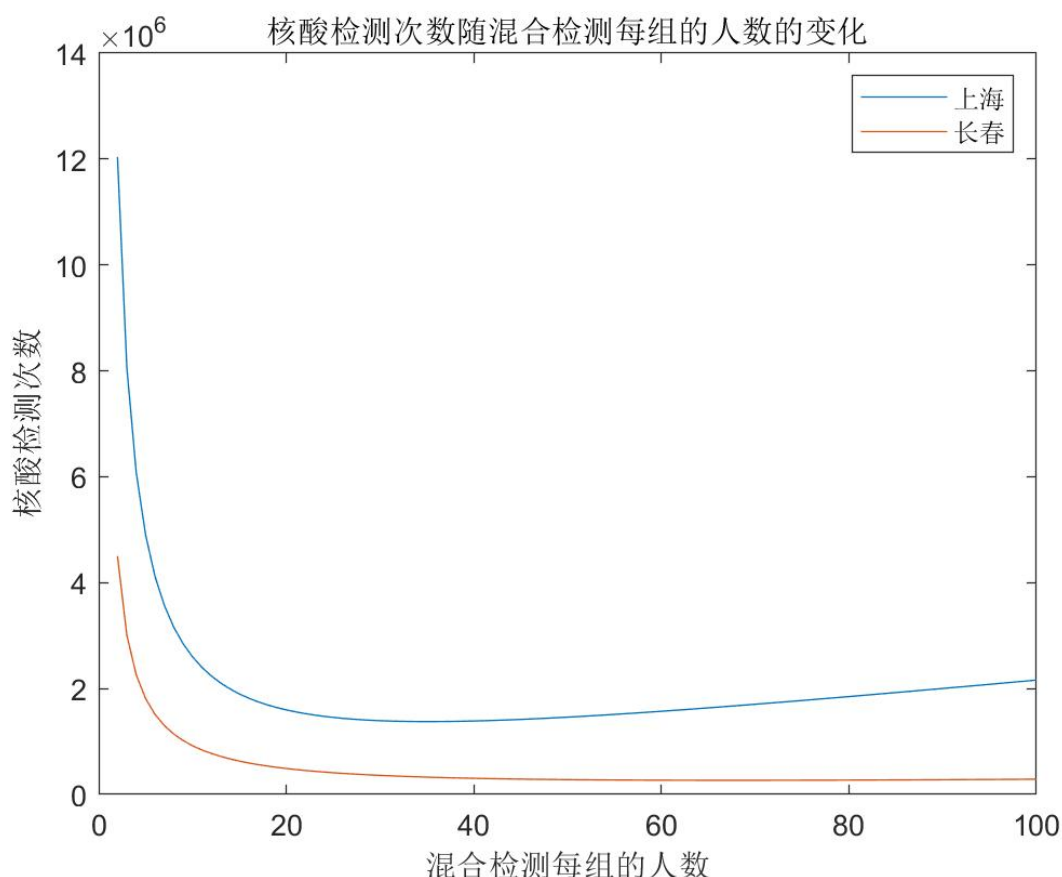


图 1: 单轮检测模型应用的实验结果

可以看到，最优的 k 值均高于实际的 k ，原因是此模型假设了核酸检测技术可以满足任意大的 k ，而在实际操作中，固定 $k = 10$ 可能较为方便。

4 多轮检测模型

4.1 问题的分析

在单轮检测模型中，不考虑在核酸检测结果发布前的新增病例以及核酸检测本身的假阴性情况。但事实上，上述两种情况都有可能发生，使得核酸检测后人群中仍有可能剩余病毒携带者，需要通过更多轮核酸检测检出。

因此，现建立多轮检测模型，除了确定混合检测每组的人数，还要确定检测的次数和间隔时间。

4.2 模型的假设

1. 病毒感染者等可能地出现在各组中；
2. 总人口数为常数；
3. 不考虑被隔离人群传播病毒的能力；
4. 核酸检测技术可以满足任意大的混合检测每组的人数。

4.3 符号的说明

符号	含义
n	总人口数
m	最初感染病毒的人口数
μ	假阴性概率
ν	假阳性概率
λ	一个病毒感染者在一轮检测的周期内将病毒传播给其他人的平均数量
θ	一个病毒感染者在一轮检测的周期内的痊愈概率
QI	当前感染病毒且被隔离的人数
QS	当前未感染病毒但被隔离的人数
NS	当前未感染病毒且未被隔离的人数
NI	当前感染病毒但未被隔离的人数
k	混合检测每组的人数

表 3: 多轮检测模型的符号说明

	感染 I	未感染 S
被隔离 Q	QI	QS
未被隔离 N	NI	NS

表 4: $QNIS$ 符号的详细说明

4.4 模型的建立

在混检的背景下，有许多可能发生的复杂情况需要考虑：

比如，单人单管检测时，出现假阴性的概率相对较大；而在混合检测时，若干位病毒感染者同时出现假阴性的概率相对较小。假阴性的不利影响被弱化。

再比如，混检结果发布之后，大量病毒感染者被隔离，无法传播病毒；同时大量健康人也被隔离等待复核，无法感染病毒。这一过程抑制了病毒的传播。

因此，很难使用解析表达式表述这一过程，转而使用算法模拟，通过参数搜索给出合适的检测次数和间隔时间。

4.4.1 初始状态

一方面，对于每个人，需要记录其是否感染、是否被隔离、是否需要复核等信息；另一方面，为了计算混合检测每组的人数 k ，以及根据一个病毒感染者单位时间将病毒传播给其他人的平均数量 λ 计算一个人被感染的概率，以及为了查看疫情情况，需要维护是否感染病毒与是否被隔离共四种状态的人数，其初始值为：

$$QI = 0$$

$$QS = 0$$

$$NS = n - m$$

$$NI = m$$

在之后的算法模拟中，需要根据模拟情况，及时调整以上各变量的值。

4.4.2 新增感染

记 λ 表示一个病毒感染者在一轮检测的周期内将病毒传播给其他人的平均数量，那么一个健康人在某一时刻感染病毒的概率为 $\lambda \frac{NI}{NS}$ 。

特别需要注意的是，由于 λ 的时间单位为一轮检测的周期， λ 事实上是一个可调整的参数。 λ 越大，表示一轮检测的周期越长； λ 越小，表示一轮检测的周期越短。为了确定检测的间隔时间，应调整 λ 的值进行多次模拟，最终得到一组合理的 λ 值，对应着一组合理的核酸检测间隔时间，使得疫情能及时有效地得到控制。

另外，使用混合检测也会对新增感染产生影响。具体而言，以混合检测结果发布、一批人被隔离等待复核为一个时间节点，以单人单管检测结果发布、一批人被解除隔离为另一个时间节点，可以将一轮检测的时间分为三部分，这三部分的时间长度与隔离情况是不同的，示意图如图 2 所示。因此在具体操作时，不仅需要根据一轮检测时间的总长度，确定 λ 的值；而且根据三个部分时间的相对长短，将 λ 分割。

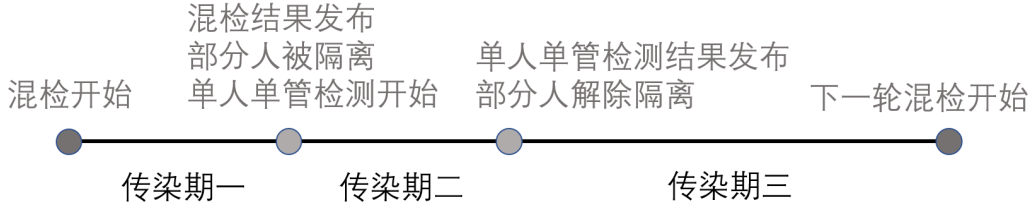


图 2: 多轮检测模型的感染期示意图

4.4.3 混合检测

混合检测前，需要决定混合检测每组的人数 k ，计算方式与单轮检测相似。不同的是，单轮检测中，不考虑假阴性与假阳性的情况，任意一个人为病毒感染者的概率为 $\frac{m}{n}$ ，也即任意一个人核酸检测结果为阳性的概率为 $\frac{m}{n}$ ；而在多轮检测中，考虑假阴性与假阳性的情况，上述概率由全概率公式改为 $\frac{m}{n}(1 - \mu) + (1 - \frac{m}{n})\nu$ 。单个人的核酸检测次数的期望改为

$$E[X] = 1 + \frac{1}{k} - (1 - p)^k$$

$$p = \frac{m}{n}(1 - \mu) + (1 - \frac{m}{n})\nu$$

需要选择 k 使上式最小。

混合检测结果发布后，需要特别注意的是，大量病毒感染者被隔离，无法传播病毒；同时大量健康人也被隔离等待复核，无法感染病毒。这一过程抑制了病毒的传播。

4.4.4 单人单管检测

单人单管检测不仅包括混合检测的复核，也是对于其他隔离人员的解除隔离判断标准。也就是说，在图 2 中，除了在混合结果发布时开始的单人单管检测，还会在下一轮混检开始前发布对其他隔离人员的单人单管检测结果，进而解除一批人的隔离。

4.4.5 痊愈

记 θ 表示一个病毒感染者在一轮检测的周期内的痊愈概率，根据这一概率，部分病毒感染者变为健康人。特别需要注意的是，和 λ 相似， θ 的时间单位也是一轮检测的周期，应随着周期的变化而变化，即随着 λ 的变化而变化。

4.5 模型的应用

本节将上述模型应用于上海市的新冠疫情情况。

与单轮检测模型一样,取总人口数 $n = 2.4 \times 10^7$,最初感染病毒的人口数 $m = 2 \times 10^4$ 。

针对新冠病毒相关参数,根据目前的研究,核酸检测的假阳性率为 0.15%-0.16%^[10],而假阴性率至少 20%^[11]。

进而得到统一参数设置如下:

参数	数值
n	2.4×10^7
m	2×10^4
μ	0.0015
ν	0.20

表 5: 多轮检测模型应用的统一参数设置

另外,可以估计我国新冠病毒的传染系数为 1 左右^[12],进而估计一个病毒感染者平均两天可以将病毒传播给一个人。若检测周期为两天,则 $\lambda = 1$;若检测周期为三天,则 $\lambda = 1.5$;若检测周期为四天,则 $\lambda = 2$ 。再加上核酸检测结果约在采样后 12 小时发布,若检测周期为两天,可以将 λ 拆分为 $0.25 + 0.25 + 0.5$,其它情况以此类推。

最后,根据目前的研究,新冠感染痊愈时间约 7-12 天^[13],进而估计一个病毒感染者每天痊愈的概率为 0.1。若检测周期为两天,则 $\theta = 0.2$;若检测周期为三天,则 $\theta = 0.3$;若检测周期为四天,则 $\theta = 0.4$ 。

进而得到可调参数设置如下:

检测周期	λ	θ
2 天	$1.0 = 0.25 + 0.25 + 0.5$	0.2
3 天	$1.5 = 0.25 + 0.25 + 1.0$	0.3
4 天	$2.0 = 0.25 + 0.25 + 1.5$	0.4

表 6: 多轮检测模型应用的可调参数设置

5 模型的评价与推广

5.1 模型的评价

5.1.1 优点

对于单轮检测模型，其给出了单个人的平均核酸检测次数的期望的解析表达式，也给出了感染病毒的人数较少时，最优混合检测每组的人数的解析表达式，既简单又直观。

对于多轮检测模型，不仅考虑了假阴性和假阳性的存在，而且考虑了在每次核酸检测之间的病毒传染与感染者痊愈情况，以及对混检结果是阳性的同组人员进行隔离的情况，非常接近实际情况。

5.1.2 缺点

对于单轮检测模型，主要考虑优化单个人的平均核酸检测次数的期望，缺少对实际情况波动性的控制。

对于多轮检测模型，一方面，由于模型考虑了假阴性、假阳性、病毒传播、感染者痊愈等等各种情况，同时现实中的人口总数非常庞大，使得模型较为复杂、计算时间长、效率低；另一方面，模型只是对各种情况进行有限次随机的算法模拟，无法涵盖所有可能，也缺少更多次的实验分析。

5.2 模型的推广

通过对问题的分析，本文的多轮混检模型在给定假阴性概率、假阳性概率、传染系数、痊愈率的基础上，使用算法模拟每个人的感染情况、隔离情况、复核情况，并统计是否感染病毒与是否被隔离四种状态的人数。该模型可以在一定区域内传染病流行时，提前估计在检测、隔离和治疗时需要投入的人力物力财力。

另外，单轮检测模型可以应用于不同传染病的初始检测试剂投放量，如学校的乙肝体检、军队入伍传染病检测等等。

参考文献

- [1] 马萍, 数学期望模型应用实例, 科技信息 (科学教研), 2007(33):458, 2007。
- [2] 上海市卫生健康委员会, 疫情通报, <https://wsjkw.sh.gov.cn/yqtb/>, 2022 年 5 月 1 日。
- [3] 上海市卫生健康委员会, 新闻发布, <https://wsjkw.sh.gov.cn/xwfb/index.html>, 2022 年 5 月 3 日。
- [4] 长春市卫生健康委员会, 长春市卫生健康委员会关于新冠肺炎疫情情况通报, <http://wjw.changchun.gov.cn/xwzx/tzgg/>, 2022 年 5 月 1 日。
- [5] 长春 e 健康, 长春市召开新型冠状病毒肺炎疫情防控工作第 27 场新闻发布会, https://mp.weixin.qq.com/s/V9W9oT-4coe_LAABoooq5g, 2022 年 5 月 2 日。
- [6] 上海市统计局, 上海市第七次全国人口普查主要数据公报 (第一号), <https://tjj.sh.gov.cn/tjgb/20210517/cc22f48611f24627bc5ee2ae96ca56d4.html>, 2022 年 5 月 1 日。
- [7] 上海市卫生健康委员会, 截至今天 8 时, 本轮全市筛查共采样 2566.5 万人, 复检完成后, 即刻转运、日清日结, 应收尽收、应隔尽隔, <https://wsjkw.sh.gov.cn/xwfb/20220405/3fa213f71e7e41bbba6942397f39b32.html>, 2022 年 5 月 2 日。
- [8] 长春市统计局, 长春市第七次全国人口普查公报 (第一号), http://tjj.changchun.gov.cn/tjgb/202106/t20210602_2830522.html, 2022 年 5 月 1 日。
- [9] 长春政事儿, 京湘助长 携春而归 李龙熙为北京、湖南援长医疗队送行, https://mp.weixin.qq.com/s/Ntb_VwW1sfeM08LOu4qttA, 2022 年 5 月 3 日。
- [10] 陈亚利, 胡明, 化学发光法检测新型冠状病毒 IgM、IgG 抗体假阳性分析, 标记免疫分析与临床, 29(02): 224-229, 2022。
- [11] Kucirka, LM; Lauer, SA; Laeyendecker, O; Lessler J; Boon, D, Variation in False-Negative Rate of Reverse Transcriptase Polymerase Chain Reaction-Based SARS-CoV-2 Tests by Time Since Exposure, Ann Intern Med, 173(4):262-267, 2020.
- [12] 红星新闻, 红星深度 | 奥密克戎传染性有多强? 美专家: “非常高”, 或为埃博拉的 5 倍, <https://www.msn.cn/zh-cn/news/national>, 2022 年 5 月 1 日。
- [13] Cristina Menni; Prof Ana M Valdes; Lorenzo Polidori; Michela Antonelli; Satya Penamakuri; Ana Nogal et al., Symptom prevalence, duration, and risk of hospital admission in individuals infected with SARS-CoV-2 during periods of omicron and delta variant dominance: a prospective observational study from the ZOE COVID Study, The Lancet, 399(10335):1618-1624, 2022.

附录

A 数据

A.1 上海市疫情数据

日期	新增确诊	新增无症状	无症状转确诊	新增感病者
3 月 1 日	1	1	0	2
3 月 2 日	3	5	0	8
3 月 3 日	2	14	0	16
3 月 4 日	3	16	0	19
3 月 5 日	0	28	0	28
3 月 6 日	3	45	0	48
3 月 7 日	4	51	0	55
3 月 8 日	3	62	0	65
3 月 9 日	4	76	0	80
3 月 10 日	11	64	0	75
3 月 11 日	5	78	0	83
3 月 12 日	1	64	0	65
3 月 13 日	41	128	2	167
3 月 14 日	9	130	0	139
3 月 15 日	5	197	0	202
3 月 16 日	8	150	1	157
3 月 17 日	57	203	0	260
3 月 18 日	8	366	0	374
3 月 19 日	17	492	6	503
3 月 20 日	24	734	0	758
3 月 21 日	31	865	0	896
3 月 22 日	4	977	0	981
3 月 23 日	4	979	0	983
3 月 24 日	29	1580	0	1609
3 月 25 日	38	2231	5	2264
3 月 26 日	45	2631	0	2676
3 月 27 日	50	3450	0	3500
3 月 28 日	96	4381	21	4456
3 月 29 日	326	5656	18	5964
3 月 30 日	355	5298	16	5637

日期	新增确诊	新增无症状	无症状转确诊	新增感病者
3 月 31 日	358	4144	20	4482
4 月 1 日	260	6051	2	6309
4 月 2 日	438	7788	73	8153
4 月 3 日	425	8581	71	8935
4 月 4 日	268	13086	4	13350
4 月 5 日	311	16766	40	17037
4 月 6 日	322	19660	15	19967
4 月 7 日	824	20398	323	20899
4 月 8 日	1015	22609	420	23204
4 月 9 日	1006	23937	191	24752
4 月 10 日	914	25173	47	26040
4 月 11 日	994	22348	273	23069
4 月 12 日	1189	25141	23	26307
4 月 13 日	2573	25146	114	27605
4 月 14 日	3200	19872	307	22765
4 月 15 日	3590	19923	922	22591
4 月 16 日	3238	21582	1177	23643
4 月 17 日	2471	19831	853	21449
4 月 18 日	3084	17332	974	19442
4 月 19 日	2494	16407	533	18368
4 月 20 日	2634	15861	459	18036
4 月 21 日	1931	15698	143	17486
4 月 22 日	2736	20634	1120	22250
4 月 23 日	1401	19657	541	20517
4 月 24 日	2472	16983	846	18609
4 月 25 日	1661	15319	968	16012
4 月 26 日	1606	11956	1253	12309
4 月 27 日	1292	9330	858	9764
4 月 28 日	5487	9545	5062	9970
4 月 29 日	1249	8932	85	9196
4 月 30 日	788	7084	683	7189

A.2 吉林省长春市疫情数据

日期	新增确诊	新增无症状	无症状转确诊	新增感病者
3 月 5 日	7	0	0	7

日期	新增确诊	新增无症状	无症状转确诊	新增感病者
3月6日	7	5	0	12
3月7日	17	6	0	23
3月8日	12	10	0	22
3月9日	23	25	0	48
3月10日	2	21	0	23
3月11日	63	97	0	160
3月12日	831	42	0	873
3月13日	430	3	22	411
3月14日	460	3	1	462
3月15日	314	3	3	314
3月16日	268	5	0	273
3月17日	595	0	0	595
3月18日	743	1	1	743
3月19日	833	3	0	836
3月20日	1079	1	2	1078
3月21日	1437	0	2	1435
3月22日	1979	0	57	1922
3月23日	1280	253	0	1533
3月24日	576	290	68	798
3月25日	554	403	7	950
3月26日	633	495	56	1072
3月27日	706	330	120	916
3月28日	622	310	11	921
3月29日	875	348	19	1204
3月30日	997	422	14	1405
3月31日	1078	444	21	1501
4月1日	1544	894	19	2419
4月2日	723	3100	25	3798
4月3日	574	2346	64	2856
4月4日	604	1336	72	1868
4月5日	817	1682	51	2448
4月6日	766	1423	50	2139
4月7日	474	1553	61	1966
4月8日	150	648	50	748
4月9日	175	703	90	788

日期	新增确诊	新增无症状	无症状转确诊	新增感病者
4 月 10 日	102	743	29	816
4 月 11 日	81	570	28	623
4 月 12 日	173	801	96	878
4 月 13 日	285	621	212	694
4 月 14 日	138	298	79	357
4 月 15 日	157	407	57	507
4 月 16 日	125	491	33	583
4 月 17 日	144	348	41	451
4 月 18 日	74	365	26	413
4 月 19 日	125	243	69	299
4 月 20 日	88	233	32	289
4 月 21 日	53	21	25	49
4 月 22 日	129	188	83	234
4 月 23 日	45	127	16	156
4 月 24 日	74	89	39	124
4 月 25 日	34	89	19	104
4 月 26 日	46	66	31	81
4 月 27 日	52	89	27	114
4 月 28 日	34	39	28	45
4 月 29 日	19	29	12	36
4 月 30 日	16	23	11	28

B 代码

B.1 单轮检测模型

```
1 clear;
2 clc;
3 close all;
4
5 % 上海
6 n = 2.4e7;
7 m = 2e4;
8
9 k = 2:100;
10 e = 1 + 1 ./ k - (1 - m / n) .^ k;
11 plot(k, n * e);
12 hold on;
13
14 % 长春
15 n = 9e6;
16 m = 2e3;
17
18 k = 2:100;
19 e = 1 + 1 ./ k - (1 - m / n) .^ k;
20 plot(k, n * e);
21
22 title('核酸检测次数随混合检测每组的人数的变化');
23 legend(['上海'; '长春'])
24 xlabel('核酸检测次数');
25 ylabel('混合检测每组的人数');
```


B.2 多轮检测模型

B.2.1 MATLAB

```
1 clear;
2 clc;
3 close all;
4
5 rng(0);
6
7 % n = 2.4e4;
8 % m = 2e1;
9 n = 2.4e7;
10 m = 2e4;
11
12 mu = 0.2; % 假阴性
13 nu = 0.0015; % 假阳性
14 lambda = [0.25, 0.25, 0.5];
15 theta = 0.2;
16 % lambda = [0.25, 0.25, 1];
17 % theta = 0.3;
18 % lambda = [0.25, 0.25, 1.5];
19 % theta = 0.4;
20
21 rounds = 60;
22 records = zeros(rounds, 3);
23
24 people = logical(sparse(n, 2));
25 % 第一个逻辑值表示是否感染病毒
26 % 第二个逻辑值表示是否被隔离
27
28 % 初始状态
29 order = randperm(n);
30 for index = 1:n
31     if order(index) <= m
32         people(index, 1) = 1;
33     end
34 end
35 QI = 0; % 感染病毒且被隔离
36 QS = 0; % 未感染病毒但被隔离
37 NS = n - m; % 未感染病毒且未被隔离
38 NI = m; % 感染病毒但未被隔离
39
40 for round = 1:rounds
41     fprintf('round %d\n', round)
```

```

42     fprintf('%8s\t%8s\t%8s\t%8s\n', 'QI', 'QS', 'NS', 'NI');
43     fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
44
45     % 新增感染
46     infect = lambda(1) * NI / NS;
47     for index = 1:n
48         if people(index, 1) == 0 && people(index, 2) == 0 && rand < infect
49             people(index, 1) = 1;
50             NI = NI + 1;
51             NS = NS - 1;
52         end
53     end
54
55     fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
56
57     % 混合检测
58     p = NI / (NI + NS);
59     p = p * (1 - mu) + (1 - p) * nu;
60     [~, k] = min(1 + 1 ./ (1:100) - (1 - p) .^ (1:100));
61     group_num = 0;
62     group_result = 0;
63     group_index = zeros(1, k);
64     recheck = logical(sparse(n, 1));
65     for index = 1:n
66         if people(index, 2) == 0
67             group_num = group_num + 1;
68             group_index(group_num) = index;
69             if people(index, 1) == 0 && rand < nu % 假阳性
70                 group_result = 1;
71             elseif people(index, 1) == 1 && rand >= mu % 真阳性
72                 group_result = 1;
73             end
74
75             if group_num == k || index == n
76                 if group_result
77                     for subindex = group_index(1:group_num)
78                         recheck(subindex) = 1;
79                         people(subindex, 2) = 1;
80                         if people(subindex, 1) == 0
81                             NS = NS - 1;
82                             QS = QS + 1;
83                         else
84                             NI = NI - 1;
85                             QI = QI + 1;
86                         end

```

```

87         end
88     end
89
90     group_num = 0;
91     group_result = 0;
92 end
93 end
94 end
95
96 fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
97
98 % 新增感染
99 infect = lambda(2) * NI / NS;
100 for index = 1:n
101     if people(index, 1) == 0 && people(index, 2) == 0 && rand < infect
102         people(index, 1) = 1;
103         NI = NI + 1;
104         NS = NS - 1;
105     end
106 end
107
108 fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
109
110 % 单人单管检测
111 for index = 1:n
112     if recheck(index) == 1
113         if people(index, 1) == 0 && rand >= nu % 真阴性
114             people(index, 2) = 0;
115             QS = QS - 1;
116             NS = NS + 1;
117         elseif people(index, 1) == 1 && rand < mu % 假阴性
118             people(index, 2) = 0;
119             QI = QI - 1;
120             NI = NI + 1;
121         end
122     end
123 end
124
125 fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
126
127 % 新增感染
128 infect = lambda(3) * NI / NS;
129 for index = 1:n
130     if people(index, 1) == 0 && people(index, 2) == 0 && rand < infect
131         people(index, 1) = 1;

```

```

132         NI = NI + 1;
133         NS = NS - 1;
134     end
135 end
136
137 fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
138
139 % 痊愈
140 for index = 1:n
141     if people(index, 1) == 1 && rand < theta
142         people(index, 1) = 0;
143         if people(index, 2) == 0
144             NI = NI - 1;
145             NS = NS + 1;
146         else
147             QI = QI - 1;
148             QS = QS + 1;
149         end
150     end
151 end
152
153 % 单人单管检测
154 for index = 1:n
155     if people(index, 2) == 1
156         if people(index, 1) == 0 % 阴性
157             if people(index, 1) == 0 && rand >= nu % 真阴性
158                 people(index, 2) = 0;
159                 QS = QS - 1;
160                 NS = NS + 1;
161             elseif people(index, 1) == 1 && rand < mu % 假阴性
162                 people(index, 2) = 0;
163                 QI = QI - 1;
164                 NI = NI + 1;
165             end
166         end
167     end
168 end
169
170 fprintf('%8d\t%8d\t%8d\t%8d\n', QI, QS, NS, NI);
171 records(round, 1) = QI;
172 records(round, 2) = NI;
173 records(round, 3) = k;
174 save('records.mat');
175 end
176

```

```

177 figure;
178 plot(records(:, 1:2));
179 title('疫情情况随时间的变化');
180 legend(["感染病毒且被隔离人数", "感染病毒但未被隔离人数"]);
181 xlabel('轮数');
182 ylabel('人数');
183
184 figure;
185 plot(records(:, 3));
186 title('混合检测每组的人数随时间的变化');
187 xlabel('轮数');
188 ylabel('混合检测每组的人数');

```

B.2.2 python

```

1 import random
2 import torch
3 import matplotlib.pyplot as plt
4
5 if __name__ == "__main__":
6     torch.manual_seed(0)
7
8     n = int(2.4e7)
9     m = int(2e4)
10    # n = int(2.4e3)
11    # m = int(2e1)
12
13    mu = 0.2
14    nu = 0.0015
15    lambd = (0.25, 0.25, 0.5)
16    theta = 0.2
17    # lambd = (0.25, 0.25, 1)
18    # theta = 0.3
19    # lambd = (0.25, 0.25, 1.5)
20    # theta = 0.4
21
22    rounds = 60
23    records = torch.zeros(rounds, 3)
24
25    # 初始状态
26    infect = set(random.choices(range(n), k=m))
27    quarantine = set()
28    QI = 0 # 感染病毒且被隔离
29    QS = 0 # 未感染病毒但被隔离
30    NS = n - m # 未感染病毒且未被隔离

```

```

31 NI = m # 感染病毒但未被隔离
32
33 for round in range(rounds):
34     print('round_{}d'.format(round))
35     print('%8s\t%8s\t%8s\t%8s' % ('QI', 'QS', 'NS', 'NI'))
36     print('%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
37
38     # 新增感染
39     p = lambd[0] * NI / NS
40     for index in range(n):
41         if index not in infect and \
42             index not in quarantine and \
43             torch.rand(1) < p:
44             infect.add(index)
45             NI += 1
46             NS -= 1
47
48     print('%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
49
50     # 混合检测
51     p = NI / (NI + NS)
52     p = p * (1 - mu) + (1 - p) * nu
53     k = torch.argmax(1 + 1 / torch.arange(1, 100) -
54                     (1 - p) ** torch.arange(1, 100))
55     group_num = 0
56     group_index = []
57     group_result = 0
58     recheck = set()
59     for index in range(n):
60         if index not in quarantine:
61             group_index.append(index)
62             if index not in infect and torch.rand(1) < nu: # 假阳性
63                 group_result = 1
64             elif index in infect and torch.rand(1) >= mu: # 真阳性
65                 group_result = 1
66             group_num += 1
67
68         if group_num == k or index == n - 1:
69             if group_result:
70                 for subindex in group_index[:group_num]:
71                     recheck.add(subindex)
72                     quarantine.add(subindex)
73                 if subindex not in infect:
74                     NS -= 1
75                     QS += 1

```

```

76         else:
77             NI -= 1
78             QI += 1
79
80             group_num = 0
81             group_result = 0
82             group_index = []
83
84     print( '%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
85
86     # 新增感染
87     p = lambd[1] * NI / NS
88     for index in range(n):
89         if index not in infect and \
90             index not in quarantine \
91             and torch.rand(1) < p:
92             infect.add(index)
93             NI += 1
94             NS -= 1
95
96     print( '%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
97
98     # 单人单管检测
99     for index in range(n):
100         if index in recheck:
101             if index not in infect and torch.rand(1) >= nu: # 真阴性
102                 quarantine.remove(index)
103                 QS -= 1
104                 NS += 1
105             elif index in infect and torch.rand(1) < mu: # 假阴性
106                 quarantine.remove(index)
107                 QI -= 1
108                 NI += 1
109
110     print( '%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
111
112     # 新增感染
113     p = lambd[2] * NI / NS
114     for index in range(n):
115         if index not in infect and \
116             index not in quarantine and \
117             torch.rand(1) < p:
118             infect.add(index)
119             NI += 1
120             NS -= 1

```

```

121
122     print( '%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
123
124     # 痊愈
125     for index in range(n):
126         if index in infect and torch.rand(1) < theta:
127             infect.remove(index)
128             if index not in quarantine:
129                 NI -= 1
130                 NS += 1
131             else:
132                 QI -= 1
133                 QS += 1
134
135     # 单人单管检测
136     for index in range(n):
137         if index in quarantine:
138             if index not in infect and torch.rand(1) >= nu: # 真阴性
139                 quarantine.remove(index)
140                 QS -= 1
141                 NS += 1
142             elif index in infect and torch.rand(1) < mu: # 假阴性
143                 quarantine.remove(index)
144                 QI -= 1
145                 NI += 1
146
147     print( '%8d\t%8d\t%8d\t%8d' % (QI, QS, NS, NI))
148     records[round, 0] = QI
149     records[round, 1] = NI
150     records[round, 2] = k
151     torch.save(records, 'records.pth')
152
153     plt.plot(records[:, :2])
154     plt.title('疫情情况随时间的变化')
155     plt.legend(["感染病毒且被隔离人数", "感染病毒但未被隔离人数"])
156     plt.xlabel('轮数')
157     plt.ylabel('人数')
158     plt.show()
159
160     plt.plot(records[:, 2])
161     plt.title('混合检测每组的人数随时间的变化')
162     plt.xlabel('轮数')
163     plt.ylabel('混合检测每组的人数')
164     plt.show()

```