五一数学建模竞赛

承 诺 书

我们仔细阅读了五一数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与本队以外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其它公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们愿意承担由此引起的一切后果。

我们授权五一数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

参赛题号（从A/B/C中选择一项填写）： A

参赛队号： T1295124844

参赛组别（研究生、本科、专科、高中）： 本科

所属学校（学校全称）： 南京财经大学

参赛队员： 队员1姓名： 侯孟敏

队员2姓名： 梅宇

队员3姓名： 侯孟慧

联系方式： Email： 1840686224@qq.com 联系电话： 18851117990

日期： 2022 年 5 月 1 日

**（除本页外不允许出现学校及个人信息）**

**五 一 数 学 建 模 竞 赛**

****

**题 目： 血管机器人的订购与生物学习**

**关键词：多阶段决策、动态规划、模拟退火算法、多目标寻优、**

**摘 要：**

本文通过分析，确定决策变量和约束条件，建立了多决策的动态规划模型，用Python代码的迭代算法和循环嵌套算法等对模型进行求解，对满足治疗所需的容器艇和操作手数量进行了分析和研究，得出题目要求的数据。

针对问题一，本文首先分析论文实现时的决策变量和约束条件，建立动态规划模型，再设计出循环嵌套算法，根据具体流程通过Python编程实现计算，代码见附录一，求解出题目中1-8周满足治疗每周所需增购的容器艇为（0,0,0,3,0,0,0,0），操作手为（14,0,0,28,0,0,0,0），最低运营成本为：7625元。

针对问题二，本文先根据题目要求，增加额外的约束条件，定义一系列变量表示1-104周容器艇和操作手的损毁数目等。基于问题一，容器艇和操作手数量减去每周损毁的数量，确定目标函数和决策变量，设计迭代算法，并根据具体流程通过Python编程实现计算，详细代码见附录二，求解出1-104周满足治疗所需的最低运营成本：667560元，其余相关结果见表6-2，最后对比分析问题一和问题二1-8周的结果数据。

针对问题三，本文先在问题二的基础上，调整约束条件：降低损毁率至10%，然后运用问题二建立的模型，同样使用迭代算法，根据流程通过Python编程实现计算，详细代码见附录四，求解出第1-104周总共需要购买的容器艇数量：498个，操作手数量：2338个，最低运营成本：439740元。其余相关结果见表7-2。

针对问题四，本文先对题目中所改变的约束条件，进行分析比对，确定条件改变后的数据变化，遵循问题三，根据新的约束条件和决策变量，利用循环嵌套算法和迭代算法，使用MATLAB编程实现计算，求解出调整后的1-104周总共购买的容器艇和操作手的数量，相关结果见表8-2**。**

针对问题五，方案一：在问题四得到的1-104周最优结果的基础上，改变约束条件，考虑最新约束下满足第105-112周需求的最佳购买方案。方案二：在问题四的约束条件下，运用相同的模型，通盘考虑第1-112周的最佳购买方案。最后运用MATLAB编程得出两个方案的最低运营成本，比较两者的差额。

**（前面两页必须使用模板格式，否则论文检测不通过）**

**此页为论文正文开始处**

1. 问题重述

随着微机电系统的发展，人类已经可以加工越来越小的机器。血管机器人的研究学习已成为国家关注的热点。这些血管机器人可以携带药物放入血管里定点治疗与血管相关的疾病，还可以充当血管清道夫，清除病毒，保持人体健康。因而，血管机器人的购买方案成为研究机构和医疗机构的重点关注问题。

请针对“血管机器人的订购与生物学习”问题，查阅资料并解决以下问题：

1. 根据所给数据，在一个熟练操作手最多“指导”10个新购买的操作手进行生物学习的条件下，考虑使运营成本达到最低，并满足1-8周治疗所需的容器艇和操作手，给出最佳的购买方案。
2. 基于问题一，鉴于血管机器人在患者血管中工作时，碰到巨噬细胞有一定风险损毁，在血管机器人每周工作损毁率达到20%的情况下，研究满足1-104周需求的最佳购买方案，将第1-8周的结果数据与问题一的第1-8周的结果数据进行对比分析，并将相关数据填入表格。
3. 基于问题二，修改以下部分条件：每名熟练操作手可以“指导”新操作手的数量调整为不超过20个；每周血管机器人的工作损毁率降低为10%。研究满足1-104周需求的最佳购买方案，并将相关数据填入表格。
4. 基于问题三，增加约束条件：购买优惠政策，研究满足问题三条件及购买优惠政策条件下，第1-104周的容器艇和操作手的调整方案，并将相关数据填入表格。
5. 基于问题四，考虑两种方案，预测105-112周的血管机器人的使用需求，并比较两种方案的第1-112周最低运营成本的差额。
6. 问题分析

问题一：本小题主要是根据1-8周具体数据，利用线性规划中的基本最优解概念使目标函数达到最优值的基本可行解，运营成本达到最低为目标函数，每周购买的操作手和容器艇数目为决策变量，每周用于训练的操作手数目为操作手购买量/10，据此写出对应算法。在Python的编译环境下，根据约束条件，得出满足1-8周治疗所需的容器艇和操作手的最佳购买方案。

问题二：本小题主要是根据1-104周的具体数据，在问题一的模型下，添加新的约束条件：血管机器人工作损毁率20%，在Python的编译环境下，利用for循环体运行代码，统计每周购买和需要保养的容器艇和操作手数量，以及每周参与训练的操作手数量和总成本。

问题三：本小题主要是基于问题二的变形求解，对问题二的模型进行局部约束条件的修改，研究第1-104周既能满足治疗，又能使运营成本达到最低所需要购买的容器艇和操作手的数目。

问题四：本小题主要是基于问题三的条件，目标函数为运营成本，决策变量与上述相同，每周购买的容器艇和操作手是多少，因此，当我们在进行问题四的建模时，只需要更改问题三的约束条件。解决方案方法与问题三相同，同时参考鲸鱼算法，以此来调整第1-104周里总共购买的容器艇和操作手的购买方案，使运营成本达到最低。

问题五：本小题主要是基于问题四的得出的结果，进行预测。在方案一的情况下：调整104周以后的容器艇和操作手价格，按照问题四的思路预测满足第105-112周的血管机器人的需求；在方案二下：按照问题四的思路，增加预测周数至112周，考虑第1-112周的血管机器人需求。最后比较两种方案的第1-112周最低运营成本的差额。

1. 模型假设
2. 所有数据均为题目所提供，具有可靠性。
3. 在所有问题中，假设一个熟练工训练后仍需要保养。
4. 符号说明

|  |  |
| --- | --- |
| 变换符号 | 符号说明 |
| arm | 机械臂 |
| container | 容器艇 |
| Initial\_arm | 初始机械臂数量 |
| Initial\_container | 初始容器艇数量 |
| weeks\_arm | 存储需要购买机械臂的周数 |
| weeks\_container | 存储需要购买容器艇的周数 |
| maintenance\_costs\_arm | 机械臂的保养费用 |
| maintenance\_costs\_container | 容器艇的保养费用 |
| add\_arm | 新增的机械臂数量 |
| add\_container | 新增的容器艇数量 |
| total\_cost | 总成本 |
| expenses\_arm | 机械臂的购买费用 |
| expenses\_container | 容器艇的购买费用 |
| training\_expenses\_arm | 训练机械臂的费用 |
| weekly\_damage\_arm | 每周损毁的机械臂数量 |
| weekly\_damage\_container | 每周损毁的容器艇数量 |
| weekly\_maintenance\_arm | 每周需要保养的机械臂数量 |
| weekly\_maintenance\_container | 每周需要保养的容器艇数量 |
| len(container)  Initial\_arm[i]  Initial\_container[i] | 周数  第i周所拥有机械臂之和  第i周所拥有容器艇之和 |
| v | 第i周参加训练的操作手数量 |

1. 问题一模型的建立和求解

5.1 基于多阶段决策中的动态规划的最优购买方案评定模型

根据题目条件所知，血管机器人的容器艇和机械手的医疗需求会随着周数的变化发生改变，基于该条件下，可知最佳购买方案的拟定为典型的多阶段决策问题。

考虑到该多阶段决策问题所求为最优解，可利用Python建立数学模型，通过变量i(周数)将多阶段决策问题巧妙化为i个非线性规划问题。

数学模型应考虑如下约束条件：

目前所处第i周小于总周数

i<Len(container)

当前第i周和下一周医疗所需操作手小于现有操作手

arm[i]+arm[i+1]>initial\_arm

当前第i周医疗所需容器艇小于现有容器艇

container[i]>initial\_container

当所处第i周不满足以上约束条件时，需根据每小题的具体条件进行add\_arm

,add\_container操作，即增加购买操作手和容器艇。最后得到最优的容器艇和操作手的购买数量和购买时间，综合统计运营成本。

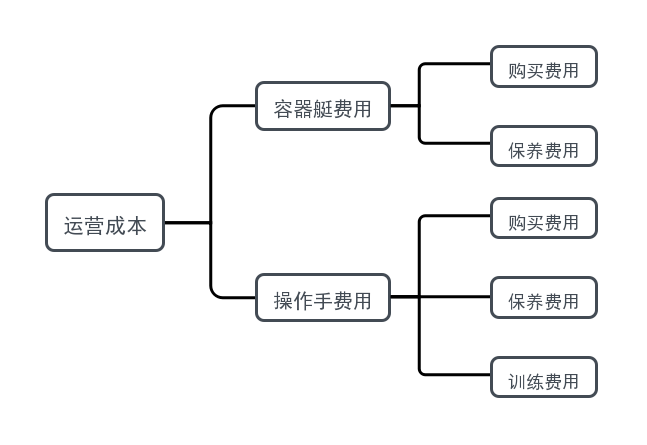


图1 运营成本的影响因素

|  |  |  |  |
| --- | --- | --- | --- |
| **表5-1血管机器人的需求（1-8周）** | | | |
| **周** | **血管机器人的使用数量** | **容器艇所需数量** | **操作手所需数量** |
| **第一周** | 11 | 11 | 44 |
| **第二周** | 5 | 5 | 20 |
| **第三周** | 4 | 4 | 16 |
| **第四周** | 7 | 7 | 28 |
| **第五周** | 16 | 16 | 64 |
| **第六周** | 6 | 6 | 24 |
| **第七周** | 5 | 5 | 20 |
| **第八周** | 7 | 7 | 28 |

5.2 容器艇和操作手购买数量的计算公式

当第i周不满足其对应的约束条件时，通过以下计算：

add\_arm[i]=arm[i]+arm[i+1]-initial\_arm （5-2-1）

add\_container[i-1]=container[i]-initial\_container （5-2-3）

将每周购买的容器艇和操作手数量存储到相应的矩阵，得到如下结果：

容器艇购买=(0 0 0 3 0 0 0 0)

机械手购买=(14 0 0 28 0 0 0 0)

**表5-2每一周的购买方案（1-8周）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **周** | container[i]**（个）** | arm[i]  **（个）** | add\_container[i]**（个）** | add\_arm[i]**（个）** |
| **第一周** | 11 | 44 | 0 | 14 |
| **第二周** | 5 | 20 | 0 | 0 |
| **第三周** | 4 | 16 | 0 | 0 |
| **第四周** | 7 | 28 | 3 | 28 |
| **第五周** | 16 | 64 | 0 | 0 |
| **第六周** | 6 | 24 | 0 | 0 |
| **第七周** | 5 | 20 | 0 | 0 |
| **第八周** | 7 | 28 | 0 | 0 |

5.3 运营成本的计算公式

增购容器艇费用expenses\_container：



(5-3-1)

增购机械臂费用expenses\_arm:



(5-3-2)

容器艇保养费用maintenance\_costs\_container:



(5-3-3)

操作手保养费用maintenance\_costs\_arm：



(5-3-4)

操作手训练费用training\_expenses\_arm：



(5-3-5)

**表4-3运营成本详细（1-8周）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **周** | **增购容器艇费用（元）** | **增购操作手费用（元）** | **容器艇保养费（元）** | **操作手保养费（元）** |  | **操作手训练费（元）（含熟练手）** | **运营成本（元）** |
| **第一周** | 0 | 140 | 20 | 20 |  | 160 | 340 |
| **第二周** | 0 | 0 | 80 | 220 |  | 0 | 300 |
| **第三周** | 0 | 0 | 90 | 240 |  | 0 | 330 |
| **第四周** | 600 | 280 | 60 | 165 |  | 310 | 1415 |
| **第五周** | 0 | 0 | 0 | 140 |  | 0 | 140 |
| **第六周** | 0 | 0 | 100 | 340 |  | 0 | 440 |
| **第七周** | 0 | 0 | 110 | 360 |  | 0 | 470 |
| **第八周** | 0 | 0 | 90 | 320 |  | 0 | 410 |
| **合计** | 600 | 4200 | 550 | 1805 |  | 470 | 7625 |

1. 问题二模型的建立和求解

6.1 基于问题一的最优购买方案评定模型

根据问题二的条件，增加约束，运用多阶段决策中的动态规划进行分析，分为以下几个步骤：第一，分析每周能够满足治疗的容器艇和操作手的数目，判断各需要增购的数目以及最合适的购买时间；第二，统计运营总成本；第三，将问题一和问题二第1-8周的结果数据进行比较分析，并完成表格。

6.2 血管机器人损毁数目的计算公式

weekly\_damage\_container[i]=[container[i]\*0.2+0.5] (6-3-1)

weekly\_damage\_arm[i]=weekly\_damage\_container[i]\*4 (6-3-2)

|  |  |  |  |
| --- | --- | --- | --- |
| **表6-1血管机器人损毁数量** | | | |
| **周** | **所需的机器人数目（个）** | **损毁的容器艇数目（个）** | **损毁的操作手数目（个）** |
| **第一周** | 11 | 2 | 8 |
| **第二周** | 5 | 1 | 4 |
| **第三周** | 4 | 1 | 4 |
| **第四周** | 7 | 1 | 4 |
| **第五周** | 16 | 3 | 12 |
| **第六周** | 6 | 1 | 4 |
| **第七周** | 5 | 1 | 4 |
| **第八周** | 7 | 1 | 4 |
| **...** | ... | ... | ... |

6.3 容器艇和操作手购买数量的计算公式

基于问题一，已知量和决策变量不发生改变，除此外，因为多增加一个损毁数量的约束条件，initial\_arm[i],initial\_container[i]在第i周结束时会发生改变

initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i]

initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i]

当第i周不满足其对应的约束条件时，通过以下计算得到第i周的购买方案：

add\_arm[i]=arm[i]+arm[i+1]-initial\_arm[i]

add\_container[i]=container[i]-initial\_container[i]

initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i]

initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i]

6.4 运营成本相关数据的计算公式

保养的操作手数量weekly\_maintenance\_arm：



(6-4-1)

保养的容器艇数量weekly\_maintenance\_container：



(6-4-2)

参加训练的操作手数量v：



(6-4-5)

增购容器艇费用expenses\_container：



(6-4-6)

增购机械臂费用expenses\_arm:



(6-4-7)

容器艇保养费用maintenance\_costs\_container:



(6-4-8)

操作手保养费用maintenance\_costs\_arm：



(6-4-9)

操作手训练费用training\_expenses\_arm：



(6-4-10)

**表6-2问题2相关结果数据**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **周次** | **购买的容器艇数量** | **购买的操作手数量** | **保养的操作手数量** | **保养的容器艇数量** | **参与训练的操作手数量（含“熟练工”和“新手”）** | **总成本**  **（单位：元）** |
| 第12周 | 5 | 20 | 26 | 1 | 22 | 3360 |
| 第26周 | 0 | 0 | 100 | 9 | 0 | 590 |
| 第52周 | 21 | 128 | 59 | 0 | 141 | 18705 |
| 第78周 | 16 | 40 | 172 | 0 | 44 | 8500 |
| 第101周 | 12 | 80 | 308 | 0 | 88 | 12820 |
| 第102周 | 17 | 36 | 344 | 0 | 40 | 9120 |
| 第103周 | 25 | 92 | 306 | 0 | 102 | 16750 |
| 第104周 | 0 | 0 | 308 | 0 | 0 | 1540 |
| 1-104周（总计） | 879 | 3826 | 13102 | 131 | 4234 | 667560 |

**表6-3运营成本详细（1-8周**）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **周** | **增购容器艇费用（元）** | **增购操作费用（元）** | **容器艇保养费（元）** | **操作手保养费（元）** | **操作手训练费用（元）（含熟练手）** | **运营成本（元）** |
| **第一周** | 0 | 1400 | 20 | 20 | 160 | 1600 |
| **第二周** | 0 | 0 | 60 | 180 | 0 | 240 |
| **第三周** | 0 | 0 | 60 | 180 | 0 | 240 |
| **第四周** | 1600 | 4400 | 20 | 75 | 490 | 6585 |
| **第五周** | 0 | 0 | 0 | 120 | 0 | 120 |
| **第六周** | 0 | 0 | 70 | 260 | 0 | 330 |
| **第七周** | 0 | 0 | 70 | 260 | 0 | 330 |
| **第八周** | 600 | 1200 | 40 | 190 | 140 | 2170 |
| **合计** | 2200 | 7000 | 340 | 1285 | 790 | 11615 |

6.5 问题一和问题二中第1-8周的结果数据对比分析

对比表5-3和表6-3，由于问题二增加了血管机器人损毁率达到20%的约束条件，所以满足问题二中治疗所需的操作手和容器艇的购买量应适当增加，因此对应的运营成本也有所增长。

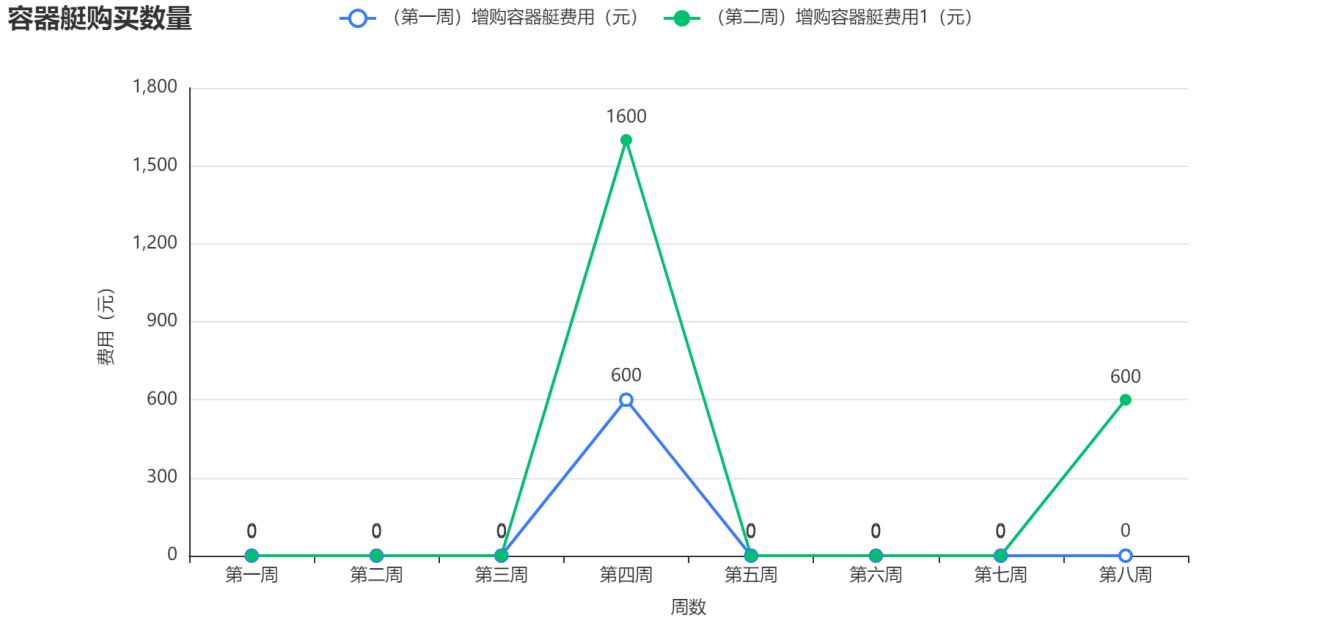


图6-1问题一和问题二第1-8周容器艇购买费用对比折线图

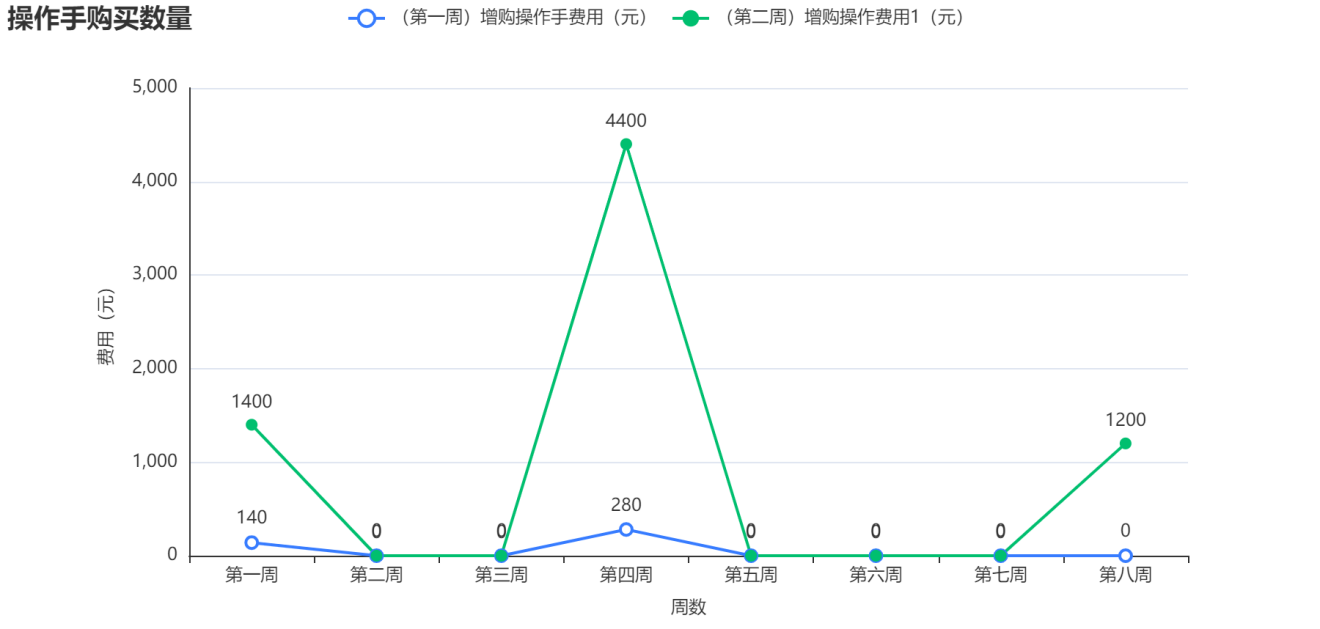


图6-2 问题一和问题二第1-8周操作手购买费用对比--折线图

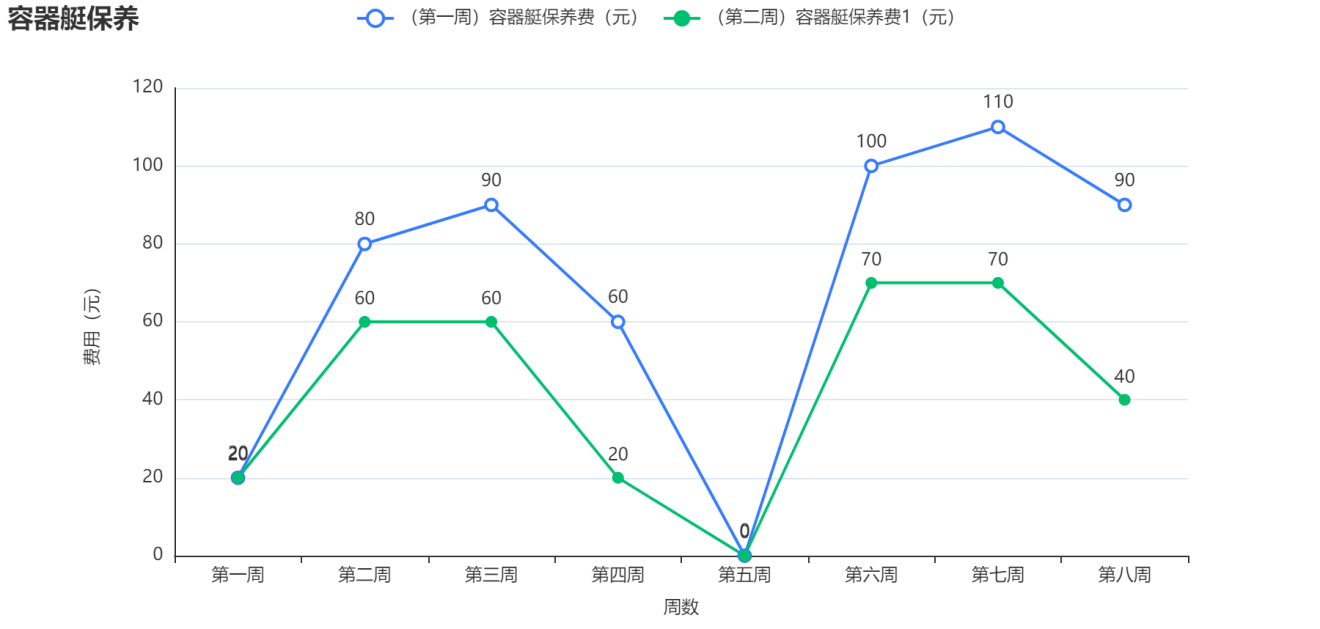


图6-3 问题一和问题二第1-8周容器艇保养费用对比--折线图

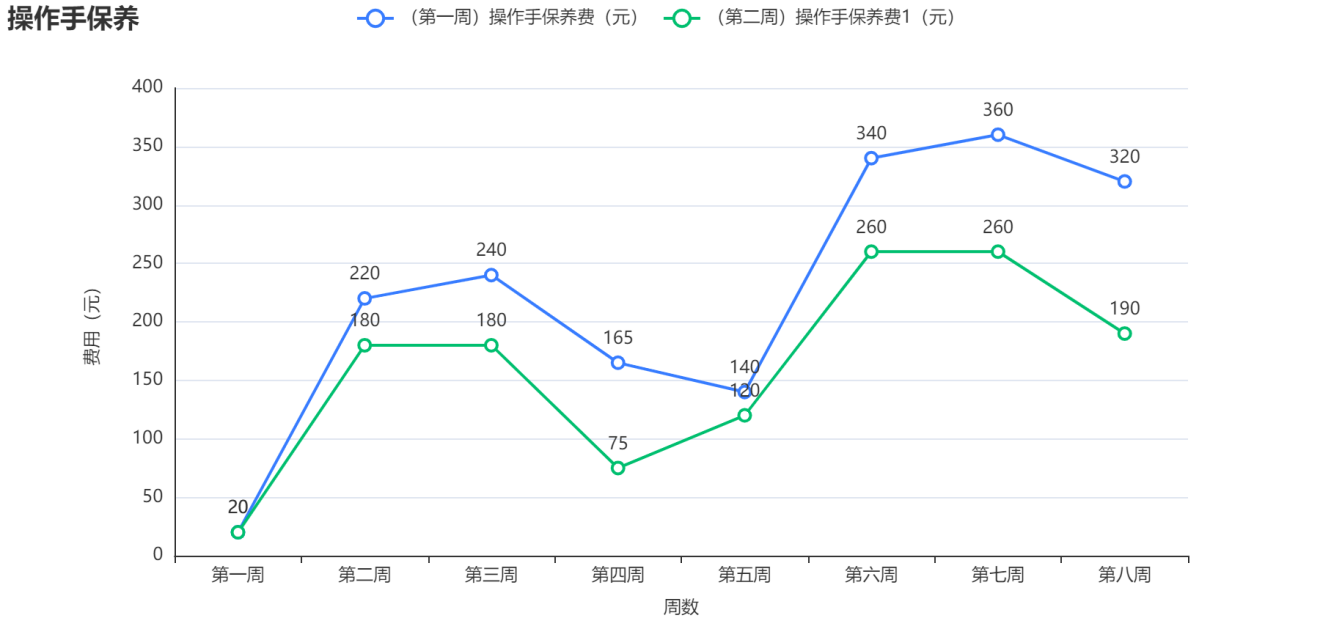


图6-4 问题一和问题二第1-8周操作手保养费用对比--折线图

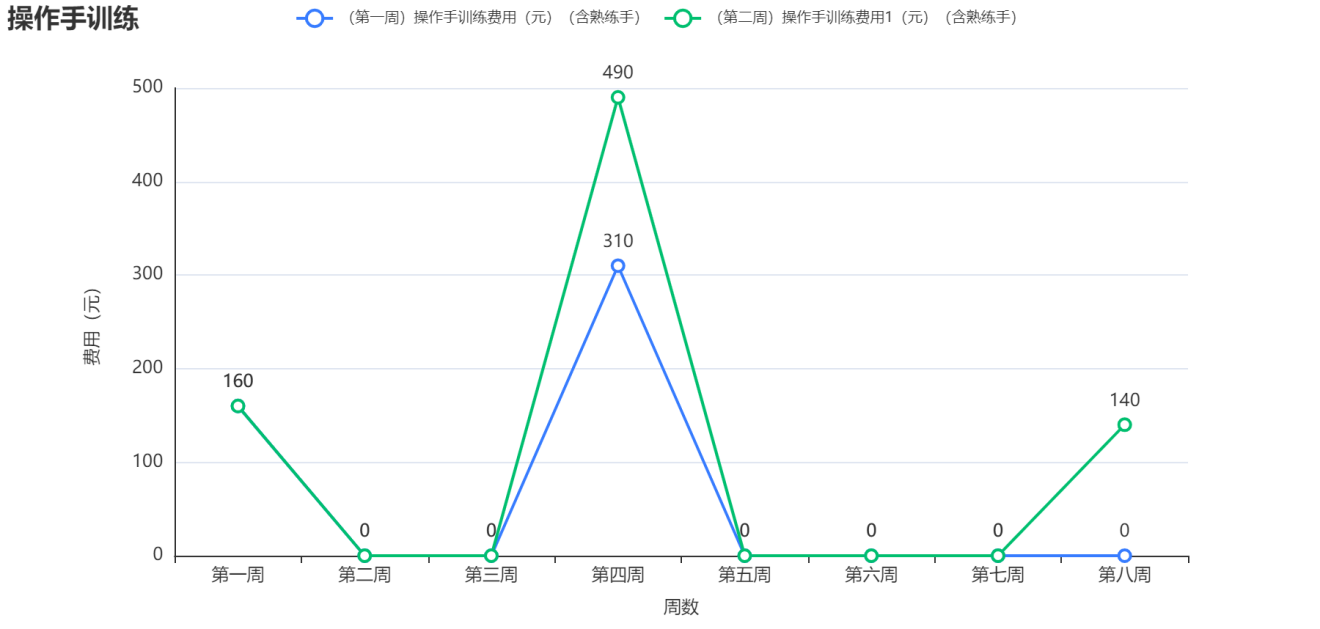


图6-5 问题一和问题二第1-8周操作手训练费用对比--折线图

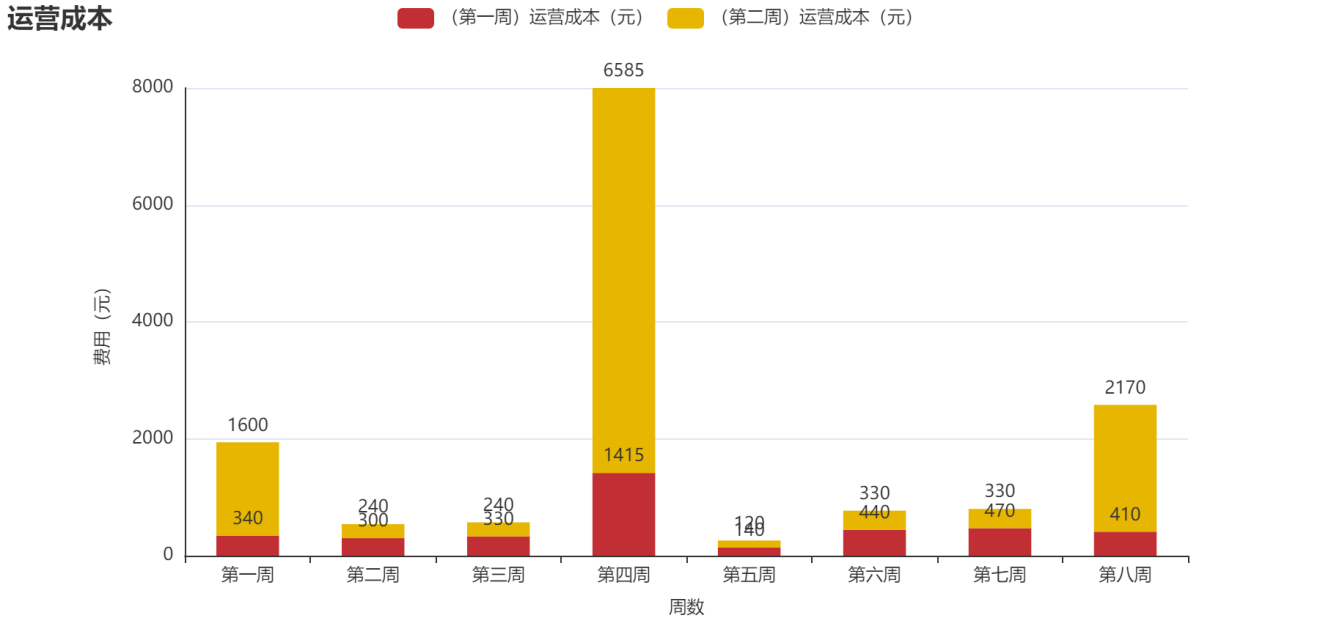


图6-6 问题一和问题二第1-8周运营成本对比--堆积柱状图

1. 问题三模型的建立和求解

7.1 基于问题二的最优购买方案评定模型

鉴于问题三本质上同问题二相同，仅仅修改了部分约束条件（损毁率降为10%，熟练手可指导的新操作手调整为不超过20），由问题二所得到的最优购买方案评定模型可不做修改，对部分参数进行修改即可得到问题三的答案。

7.2 血管机器人损毁数目的计算公式

|  |  |  |
| --- | --- | --- |
|  | weekly\_damage\_container[i]=[container[i]\*0.1+0.5]  weekly\_damage\_arm[i]=weekly\_damage\_container[i]\*4 | (7-2-2)  (7-2-1) |

7.3 容器艇和操作手购买数量的计算公式

基于问题二，没有新的约束条件，仅仅修改损毁率和熟练手指导的新操作手数量，因而可以套用问题二的计算公式。

|  |  |  |
| --- | --- | --- |
|  | initial\_arm[i],initial\_container[i] | (7-3-1) |

在第i周结束时会发生改变

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i] | | | (7-3-2) |
|  | | initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i] | (7-3-3) | |

当第i周不满足其对应的约束条件时，通过以下计算得到第i周的购买方案：

|  |  |  |
| --- | --- | --- |
|  | add\_arm[i]=arm[i]+arm[i+1]-initial\_arm[i]  add\_container[i]=container[i]-initial\_container[i]  initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i]  initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i] | (7-3-4)  (7-3-5)  (7-3-6)  (7-3-7) |

7.4 运营成本相关数据的计算公式

保养的操作手数量weekly\_maintenance\_arm：

|  |  |  |
| --- | --- | --- |
|  |  | (7-4-1) |

保养的容器艇数量weekly\_maintenance\_container：



(7-4-2)

参加训练的操作手数量v：



(7-4-3)

增购容器艇费用expenses\_container：

IMG_256

(7-4-4)

增购机械臂费用expenses\_arm:



(7-4-5)

容器艇保养费用maintenance\_costs\_container:



(7-4-6)

操作手保养费用maintenance\_costs\_arm：



(7-4-7)

操作手训练费用training\_expenses\_arm：



(7-4-8)

表7-2问题3相关结果数据

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **周次** | **购买的容器艇数量** | **购买的操作手数量** | **保养的操作手数量** | **保养的容器艇数量** | **参与训练的操作手数量（含“熟练工”和“新手”）** | **总成本**  **（单位：元）** |
| 第12周 | 3 | 12 | 35 | 3 | 3 | 2135 |
| 第26周 | 0 | 0 | 116 | 11 | 0 | 690 |
| 第52周 | 18 | 116 | 78 | 0 | 122 | 16810 |
| 第78周 | 10 | 20 | 195 | 1 | 21 | 5195 |
| 第101周 | 1 | 40 | 354 | 0 | 42 | 6390 |
| 第102周 | 7 | 0 | 392 | 0 | 0 | 3360 |
| 第103周 | 16 | 44 | 361 | 0 | 47 | 9875 |
| 第104周 | 0 | 0 | 344 | 0 | 0 | 1720 |
| 1-104周（总计） | 498 | 2338 | 15758 | 281 | 2474 | 439740 |



图7-1问题三周数和总成本的关系折线图

1. 问题四模型的建立和求解

8.1 基于问题三的最优购买方案评定模型

问题四在问题三的基础上，改变了优惠政策的约束条件，目标函数和已知量不变，以同样的模型，对1-104周总共购买的容器艇和操作手数量进行调整，并完成相关表格。

8.2 优惠价格相关数据的计算公式

1. 随机生成一个初始值，并产生一个初始变化数量（容器艇），求出运营成本因此省下的钱ΔPr
2. 若ΔPk≤0则变化数量生效并继续产生位移再判断；若ΔPr > 0则以值为Pk的概率来判断是否生效
3. Pk公式，式中c为购买容器艇时，每个容器艇节省的钱

|  |  |  |
| --- | --- | --- |
|  |  | (8-2-1) |

4. 转第1步继续执行，直到达到平衡状态为止

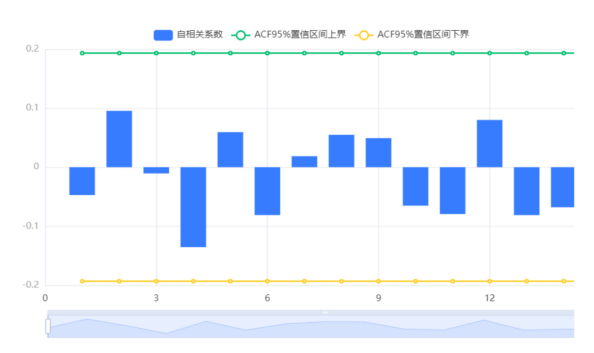
1. 问题五模型的建立和求解

9.1 基于问题四的最优购买方案评定模型

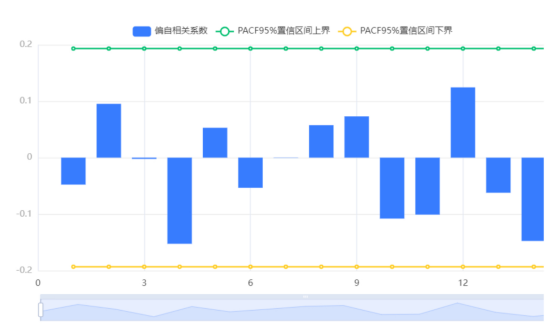
# 本题预比较方案一，方案二最低运营成本差额，应先进行时间序列分析，得到105-112周血管机器人的使用需求。自回归移动平均模型是目前最常用的拟合平稳序列的模型，本题具体采用ARMA模型进行时间序列分析预测。

通过对自相关系数（ACF）和偏自相关系数（PACF）进行分析:

模型残差自相关图（ACF）



模型残差偏自相关图（PACF）



得到下列数学模型

|  |  |  |
| --- | --- | --- |
|  | y(t)=0.898+0.125\*y(t-1)-0.209\*y(t-2)-0.27\*y(t-3)+0.515\*y(t-4)-0.272\*y(t-5) | (9-1-1) |

从而得到105-112周血管机器人使用需求

|  |  |
| --- | --- |
| 时间（周） | 使用需求 |
| 105 | 111 |
| 106 | 105 |
| 107 | 103 |
| 108 | 106 |
| 109 | 112 |
| 110 | 109 |
| 111 | 107 |
| 112 | 110 |

1. 模型评价

在问题二和问题三中，因为题目要求每周的血管机器人损毁数目按四舍五入取整，在Python的编译环境下，四舍五入取整函数为round()，但是round(0.5)=0而不是1，这就导致我们问题三的容器艇总共多买了14个，操作手多买了56个，总成本多了7715元。所以我们使用了int(损毁的数量+0.5)来进行四舍五入。

1. 参考文献
2. <https://www.jianshu.com/p/3586a6e0e246>
3. <https://blog.csdn.net/weixin_44613063/article/details/104151489>

**附录：**

**附录一：问题一代码**

import math

arm = [44, 20, 16, 28, 64, 24, 20, 28]

container = [11, 5, 4, 7, 16, 6, 5, 7]

initial\_arm = 50

initial\_container = 13

weeks\_arm = []

weeks\_container = []

maintenance\_costs\_arm, maintenance\_costs\_container = 0, 0

num1 = 0

num2 = 0

add\_arm = []

add\_container = []

total\_cost = 0

for i in range(len(arm)):

if (i + 1 < len(arm)):

s = arm[i] + arm[i + 1]

if (s > initial\_arm):

num1 = s - initial\_arm

add\_arm.append(num1)

initial\_arm = s

weeks\_arm.append(i)

if (container[i] > initial\_container):

num2 = container[i] - initial\_container

add\_container.append(num2)

initial\_container = container[i]

weeks\_container.append(i - 1)

maintenance\_costs\_arm = maintenance\_costs\_arm + (initial\_arm - arm[i]) \* 5

maintenance\_costs\_container = maintenance\_costs\_container + (initial\_container - container[i]) \* 10

maintenance\_costs\_arm = maintenance\_costs\_arm - sum(add\_arm) \* 5

for i in range(len(weeks\_arm)):

print("第", weeks\_arm[i] + 1, "周购买的机械臂数量为：", add\_arm[i])

for i in range(len(weeks\_container)):

print("第", weeks\_container[i] + 1, "周购买的容器艇数量为：", add\_container[i])

expenses\_arm = (initial\_arm - 50) \* 100

expenses\_container = (initial\_container - 13) \* 200

training\_expenses\_arm = (initial\_arm - 50) \* 10 + math.ceil(sum(add\_arm) / 10) \* 5

total\_cost = expenses\_arm + expenses\_container + training\_expenses\_arm + maintenance\_costs\_arm + maintenance\_costs\_container

print("总成本：", total\_cost, "元")

**附录二：问题二代码**

import math

from openpyxl import Workbook

from openpyxl import load\_workbook

from openpyxl.utils import get\_column\_letter

container = [11, 5, 4, 7, 16, 6, 5, 7,

13, 6, 5, 7, 12, 5, 4, 6,

9, 5, 5, 11, 29, 21, 17, 20,

27, 13, 9, 10, 16, 6, 5, 7,

11, 5, 5, 6, 12, 7, 7, 10,

15, 10, 9, 11, 15, 10, 10, 16,

26, 21, 23, 36, 50, 45, 45, 49,

57, 43, 40, 44, 52, 43, 42, 45,

52, 41, 39, 41, 48, 35, 34, 35,

42, 34, 36, 43, 55, 48, 54, 65,

80, 70, 74, 85, 101, 89, 88, 90,

100, 87, 88, 89, 104, 89, 89, 90,

106, 96, 94, 99, 109, 99, 96, 102]

arm = []

for i in range(len(container)):

arm.append(container[i] \* 4)

initial\_arm = 50

initial\_container = 13

weeks\_arm = []

weeks\_container = []

maintenance\_costs\_arm, maintenance\_costs\_container = 0, 0

num1 = 0

num2 = 0

add\_arm = []

add\_container = []

weekly\_damage\_arm = []

weekly\_damage\_container = []

existing\_arms = []

weekly\_maintenance\_arm = []

weekly\_maintenance\_container = []

v = []

total\_cost = 0

a = 0

for i in range(len(container)):

weekly\_damage\_container.append(int (container[i] \* 0.2 + 0.5))

weekly\_damage\_arm.append(4 \* weekly\_damage\_container[i])

if (i + 1 < len(arm)):

s = arm[i] + arm[i + 1]

if (s > initial\_arm):

num1 = s - initial\_arm

add\_arm.append(num1)

a = math.ceil(num1 / 10)

v.append(num1 + a)

weekly\_maintenance\_arm.append(initial\_arm - arm[i] - a)

initial\_arm = s

weeks\_arm.append(i)

else:

add\_arm.append(0)

v.append(0)

weekly\_maintenance\_arm.append(initial\_arm - arm[i])

if (container[i] > initial\_container):

num2 = container[i] - initial\_container

add\_container.append(num2)

initial\_container = container[i]

weeks\_container.append(i - 1)

else:

add\_container.append(0)

weekly\_maintenance\_container.append(initial\_container - container[i])

initial\_arm = initial\_arm - weekly\_damage\_arm[i]

if (i + 1 >= len(container)):

break

else:

maintenance\_costs\_arm = maintenance\_costs\_arm + weekly\_maintenance\_arm[i] \* 5

maintenance\_costs\_container = maintenance\_costs\_container + (initial\_container - container[i]) \* 10

initial\_container = initial\_container - weekly\_damage\_container[i]

existing\_arms.append(initial\_arm - add\_arm[i] + weekly\_damage\_arm[i])

#另长度为104

add\_arm.append(0)

add\_container.remove(add\_container[0])

add\_container.append(0)

v.append(0)

existing\_arms.append(existing\_arms[102]-weekly\_damage\_arm[102]+add\_arm[102])

weekly\_maintenance\_arm.append(existing\_arms[len(existing\_arms)-1] - arm[len(arm)-1])

maintenance\_costs\_arm = maintenance\_costs\_arm + weekly\_maintenance\_arm[len(weekly\_maintenance\_arm)-1] \* 5

for i in range(len(add\_arm)):

print("第", i + 1, "周购买的操作手数量为：", add\_arm[i])

for i in range(len(add\_container)):

print("第", i+1, "周购买的容器艇数量为：", add\_container[i])

total\_add\_arm = sum(add\_arm)

expenses\_arm = sum(add\_arm) \* 100

expenses\_container = sum(add\_container) \* 200

training\_expenses\_arm = sum(v) \* 10

total\_cost = expenses\_arm + expenses\_container + training\_expenses\_arm + maintenance\_costs\_arm + maintenance\_costs\_container

print("总共购买的操作手数量为：%d个" % sum(add\_arm))

print("总共购买的容器艇数量为：%d个"% sum(add\_container))

print("总成本：%d元"% total\_cost)

#周数（104周）

weeks = []

for i in range(0, 104):

weeks1 = "第%d周"% (i+1)

weeks.append(weeks1)

#统计每周总成本

weeks\_cost = []

for i in range(0, 104):

weeks\_cost.append(add\_container[i]\*200+add\_arm[i]\*100+weekly\_maintenance\_arm[i]\*5+weekly\_maintenance\_container[i]\*10+v[i]\*10)

# 绘制表格

wb = Workbook()

sheet = wb.active

sheet.title='问题二：1-104周的结果数据'

sheet.append(['周数', '购买的容器艇数量','现有的操作手数量', '购买的操作手数量','保养的操作手数量','保养的容器艇数量','参与训练的操作手数量','总成本（单位：元）'])

for i in range(0,104):

sheet.append([weeks[i],add\_container[i],existing\_arms[i],add\_arm[i],weekly\_maintenance\_arm[i],weekly\_maintenance\_container[i],v[i],weeks\_cost[i]])

sheet.append(["1-104周总和",sum(add\_container),sum(existing\_arms),sum(add\_arm),sum(weekly\_maintenance\_arm),sum(weekly\_maintenance\_container),sum(v),sum(weeks\_cost)])

wb.save('问题二：1-104周的结果数据.xlsx')

wb = load\_workbook('问题二：1-104周的结果数据.xlsx')

sheet = wb.active

#表格列宽自适应

lks = []

for i in range(1, sheet.max\_column + 1):

lk = 1

for j in range(1, sheet.max\_row + 1):

sz = sheet.cell(row=j, column=i).value

if isinstance(sz, str):

lk1 = len(sz.encode('gbk'))

else:

lk1 = len(str(sz))

if lk < lk1:

lk = lk1

# print(lk)

lks.append(lk)

# 第二步：设置列宽

for i in range(1, sheet.max\_column + 1):

k = get\_column\_letter(i)

sheet.column\_dimensions[k].width = lks[i - 1] + 2

wb.close()

wb.save('问题二：1-104周的结果数据.xlsx')

**附录三：问题二代码生成的表格**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 周数 | 购买的容器艇数量 | 现有的操作手数量 | 购买的操作手数量 | 保养的操作手数量 | 保养的容器艇数量 | 参与训练的操作手数量 | 总成本（单位：元） |
| 第1周 | 0 | 50 | 14 | 4 | 2 | 16 | 1600 |
| 第2周 | 0 | 56 | 0 | 36 | 6 | 0 | 240 |
| 第3周 | 0 | 52 | 0 | 36 | 6 | 0 | 240 |
| 第4周 | 8 | 48 | 44 | 15 | 2 | 49 | 6585 |
| 第5周 | 0 | 88 | 0 | 24 | 0 | 0 | 120 |
| 第6周 | 0 | 76 | 0 | 52 | 7 | 0 | 330 |
| 第7周 | 0 | 72 | 0 | 52 | 7 | 0 | 330 |
| 第8周 | 3 | 68 | 12 | 38 | 4 | 14 | 2170 |
| 第9周 | 0 | 76 | 0 | 24 | 0 | 0 | 120 |
| 第10周 | 0 | 64 | 0 | 40 | 4 | 0 | 240 |
| 第11周 | 0 | 60 | 0 | 40 | 4 | 0 | 240 |
| 第12周 | 5 | 56 | 20 | 26 | 1 | 22 | 3360 |
| 第13周 | 0 | 72 | 0 | 24 | 0 | 0 | 120 |
| 第14周 | 0 | 64 | 0 | 44 | 5 | 0 | 270 |
| 第15周 | 0 | 60 | 0 | 44 | 5 | 0 | 270 |
| 第16周 | 2 | 56 | 4 | 31 | 2 | 5 | 1025 |
| 第17周 | 0 | 56 | 0 | 20 | 0 | 0 | 100 |
| 第18周 | 0 | 48 | 0 | 28 | 2 | 0 | 160 |
| 第19周 | 6 | 44 | 20 | 22 | 1 | 22 | 3540 |
| 第20周 | 20 | 60 | 100 | 6 | 0 | 110 | 15130 |
| 第21周 | 0 | 152 | 48 | 31 | 0 | 53 | 5485 |
| 第22周 | 0 | 176 | 0 | 92 | 2 | 0 | 480 |
| 第23周 | 4 | 160 | 0 | 92 | 2 | 0 | 1280 |
| 第24周 | 11 | 148 | 40 | 64 | 0 | 44 | 6960 |
| 第25周 | 0 | 172 | 0 | 64 | 0 | 0 | 320 |
| 第26周 | 0 | 152 | 0 | 100 | 9 | 0 | 590 |
| 第27周 | 0 | 140 | 0 | 104 | 10 | 0 | 620 |
| 第28周 | 1 | 132 | 0 | 92 | 7 | 0 | 730 |
| 第29周 | 0 | 124 | 0 | 60 | 0 | 0 | 300 |
| 第30周 | 0 | 112 | 0 | 88 | 7 | 0 | 510 |
| 第31周 | 0 | 108 | 0 | 88 | 7 | 0 | 510 |
| 第32周 | 1 | 104 | 0 | 76 | 4 | 0 | 620 |
| 第33周 | 0 | 100 | 0 | 56 | 0 | 0 | 280 |
| 第34周 | 0 | 92 | 0 | 72 | 4 | 0 | 400 |
| 第35周 | 0 | 88 | 0 | 68 | 3 | 0 | 370 |
| 第36周 | 6 | 84 | 0 | 60 | 1 | 0 | 1510 |
| 第37周 | 0 | 80 | 0 | 32 | 0 | 0 | 160 |
| 第38周 | 0 | 72 | 0 | 44 | 3 | 0 | 250 |
| 第39周 | 2 | 68 | 0 | 40 | 2 | 0 | 620 |
| 第40周 | 7 | 64 | 36 | 20 | 0 | 40 | 5500 |
| 第41周 | 0 | 92 | 8 | 31 | 0 | 9 | 1045 |
| 第42周 | 0 | 88 | 0 | 48 | 2 | 0 | 260 |
| 第43周 | 3 | 80 | 0 | 44 | 1 | 0 | 830 |
| 第44周 | 6 | 72 | 32 | 24 | 0 | 36 | 4880 |
| 第45周 | 0 | 96 | 4 | 35 | 0 | 5 | 625 |
| 第46周 | 0 | 88 | 0 | 48 | 2 | 0 | 260 |
| 第47周 | 8 | 80 | 24 | 37 | 0 | 27 | 4455 |
| 第48周 | 13 | 96 | 72 | 24 | 0 | 80 | 10720 |
| 第49周 | 0 | 156 | 32 | 48 | 0 | 36 | 3800 |
| 第50周 | 6 | 168 | 8 | 83 | 0 | 9 | 2505 |
| 第51周 | 18 | 160 | 76 | 60 | 0 | 84 | 12340 |
| 第52周 | 21 | 216 | 128 | 59 | 0 | 141 | 18705 |
| 第53周 | 5 | 316 | 64 | 109 | 0 | 71 | 8655 |
| 第54周 | 9 | 340 | 20 | 158 | 0 | 22 | 4810 |
| 第55周 | 13 | 324 | 52 | 138 | 0 | 58 | 9070 |
| 第56周 | 18 | 340 | 84 | 135 | 0 | 93 | 13605 |
| 第57周 | 0 | 384 | 16 | 154 | 0 | 18 | 2550 |
| 第58周 | 3 | 356 | 0 | 184 | 3 | 0 | 1550 |
| 第59周 | 12 | 320 | 16 | 158 | 0 | 18 | 4970 |
| 第60周 | 17 | 304 | 80 | 120 | 0 | 88 | 12880 |
| 第61周 | 1 | 348 | 32 | 136 | 0 | 36 | 4440 |
| 第62周 | 8 | 340 | 0 | 168 | 0 | 0 | 2440 |
| 第63周 | 11 | 304 | 44 | 131 | 0 | 49 | 7745 |
| 第64周 | 16 | 316 | 72 | 128 | 0 | 80 | 11840 |
| 第65周 | 0 | 352 | 20 | 142 | 0 | 22 | 2930 |
| 第66周 | 5 | 332 | 0 | 168 | 1 | 0 | 1850 |
| 第67周 | 10 | 300 | 20 | 142 | 0 | 22 | 4930 |
| 第68周 | 15 | 288 | 68 | 117 | 0 | 75 | 11135 |
| 第69周 | 0 | 324 | 8 | 131 | 0 | 9 | 1545 |
| 第70周 | 3 | 292 | 0 | 152 | 3 | 0 | 1390 |
| 第71周 | 8 | 264 | 12 | 126 | 0 | 14 | 3570 |
| 第72周 | 14 | 248 | 60 | 102 | 0 | 66 | 9970 |
| 第73周 | 0 | 280 | 24 | 109 | 0 | 27 | 3215 |
| 第74周 | 9 | 272 | 8 | 135 | 0 | 9 | 3365 |
| 第75周 | 14 | 252 | 64 | 101 | 0 | 71 | 10415 |
| 第76周 | 21 | 288 | 104 | 105 | 0 | 115 | 16275 |
| 第77周 | 4 | 356 | 56 | 130 | 0 | 62 | 7670 |
| 第78周 | 16 | 368 | 40 | 172 | 0 | 44 | 8500 |
| 第79周 | 22 | 368 | 108 | 141 | 0 | 119 | 17095 |
| 第80周 | 28 | 432 | 148 | 157 | 0 | 163 | 22815 |
| 第81周 | 6 | 528 | 72 | 200 | 0 | 80 | 10200 |
| 第82周 | 18 | 536 | 40 | 252 | 0 | 44 | 9300 |
| 第83周 | 26 | 520 | 116 | 212 | 0 | 128 | 19140 |
| 第84周 | 33 | 576 | 168 | 219 | 0 | 185 | 26345 |
| 第85周 | 8 | 676 | 84 | 263 | 0 | 93 | 12245 |
| 第86周 | 17 | 680 | 28 | 321 | 0 | 31 | 8115 |
| 第87周 | 20 | 636 | 76 | 276 | 0 | 84 | 13820 |
| 第88周 | 28 | 640 | 120 | 268 | 0 | 132 | 20260 |
| 第89周 | 7 | 688 | 60 | 282 | 0 | 66 | 9470 |
| 第90周 | 18 | 668 | 32 | 316 | 0 | 36 | 8740 |
| 第91周 | 19 | 632 | 76 | 272 | 0 | 84 | 13600 |
| 第92周 | 33 | 636 | 136 | 266 | 0 | 150 | 23030 |
| 第93周 | 6 | 700 | 72 | 276 | 0 | 80 | 10580 |
| 第94周 | 18 | 688 | 24 | 329 | 0 | 27 | 7915 |
| 第95周 | 19 | 640 | 76 | 276 | 0 | 84 | 13620 |
| 第96周 | 34 | 644 | 140 | 270 | 0 | 154 | 23690 |
| 第97周 | 11 | 712 | 96 | 278 | 0 | 106 | 14250 |
| 第98周 | 17 | 724 | 36 | 336 | 0 | 40 | 9080 |
| 第99周 | 24 | 684 | 88 | 299 | 0 | 97 | 16065 |
| 第100周 | 30 | 696 | 136 | 286 | 0 | 150 | 22530 |
| 第101周 | 12 | 752 | 80 | 308 | 0 | 88 | 12820 |
| 第102周 | 17 | 744 | 36 | 344 | 0 | 40 | 9120 |
| 第103周 | 25 | 700 | 92 | 306 | 0 | 102 | 16750 |
| 第104周 | 0 | 716 | 0 | 308 | 0 | 0 | 1540 |
| 1-104周总和 | 879 | 29770 | 3826 | 13102 | 131 | 4234 | 667560 |

**附录四：问题三代码**

import math

from openpyxl import Workbook

from openpyxl import load\_workbook

from openpyxl.utils import get\_column\_letter

container = [11, 5, 4, 7, 16, 6, 5, 7,

13, 6, 5, 7, 12, 5, 4, 6,

9, 5, 5, 11, 29, 21, 17, 20,

27, 13, 9, 10, 16, 6, 5, 7,

11, 5, 5, 6, 12, 7, 7, 10,

15, 10, 9, 11, 15, 10, 10, 16,

26, 21, 23, 36, 50, 45, 45, 49,

57, 43, 40, 44, 52, 43, 42, 45,

52, 41, 39, 41, 48, 35, 34, 35,

42, 34, 36, 43, 55, 48, 54, 65,

80, 70, 74, 85, 101, 89, 88, 90,

100, 87, 88, 89, 104, 89, 89, 90,

106, 96, 94, 99, 109, 99, 96, 102]

arm = []

for i in range(len(container)):

arm.append(container[i] \* 4)

initial\_arm = 50

initial\_container = 13

weeks\_arm = []

weeks\_container = []

maintenance\_costs\_arm, maintenance\_costs\_container = 0, 0

num1 = 0

num2 = 0

add\_arm = []

add\_container = []

weekly\_damage\_arm = []

weekly\_damage\_container = []

existing\_arms = []

weekly\_maintenance\_arm = []

weekly\_maintenance\_container = []

v = []

total\_cost = 0

a = 0

for i in range(len(container)):

weekly\_damage\_container.append(int (container[i] \* 0.1 + 0.5))

weekly\_damage\_arm.append(4 \* weekly\_damage\_container[i])

if (i + 1 < len(arm)):

s = arm[i] + arm[i + 1]

if (s > initial\_arm):

num1 = s - initial\_arm

add\_arm.append(num1)

a = math.ceil(num1 / 20)

v.append(num1 + a)

weekly\_maintenance\_arm.append(initial\_arm - arm[i] - a)

initial\_arm = s

weeks\_arm.append(i)

else:

add\_arm.append(0)

v.append(0)

weekly\_maintenance\_arm.append(initial\_arm - arm[i])

if (container[i] > initial\_container):

num2 = container[i] - initial\_container

add\_container.append(num2)

initial\_container = container[i]

weeks\_container.append(i - 1)

else:

add\_container.append(0)

weekly\_maintenance\_container.append(initial\_container - container[i])

initial\_arm = initial\_arm - weekly\_damage\_arm[i]

if (i + 1 >= len(container)):

break

else:

maintenance\_costs\_arm = maintenance\_costs\_arm + weekly\_maintenance\_arm[i] \* 5

maintenance\_costs\_container = maintenance\_costs\_container + (initial\_container - container[i]) \* 10

initial\_container = initial\_container - weekly\_damage\_container[i]

existing\_arms.append(initial\_arm - add\_arm[i] + weekly\_damage\_arm[i])

add\_arm.append(0)

add\_container.remove(add\_container[0])

add\_container.append(0)

v.append(0)

existing\_arms.append(existing\_arms[102]-weekly\_damage\_arm[102]+add\_arm[102])

weekly\_maintenance\_arm.append(existing\_arms[len(existing\_arms)-1] - arm[len(arm)-1])

maintenance\_costs\_arm = maintenance\_costs\_arm + weekly\_maintenance\_arm[len(weekly\_maintenance\_arm)-1] \* 5

for i in range(len(add\_arm)):

print("第", i + 1, "周购买的操作手数量为：", add\_arm[i])

for i in range(len(add\_container)):

print("第", i+1, "周购买的容器艇数量为：", add\_container[i])

total\_add\_arm = sum(add\_arm)

expenses\_arm = sum(add\_arm) \* 100

expenses\_container = sum(add\_container) \* 200

training\_expenses\_arm = sum(v) \* 10

total\_cost = expenses\_arm + expenses\_container + training\_expenses\_arm + maintenance\_costs\_arm + maintenance\_costs\_container

print("总共购买的操作手数量为：%d个" % sum(add\_arm))

print("总共购买的容器艇数量为：%d个"% sum(add\_container))

print("总成本：%d元"% total\_cost)

#周数（104周）

weeks = []

for i in range(0, 104):

weeks1 = "第%d周"% (i+1)

weeks.append(weeks1)

#统计每周总成本

weeks\_cost = []

for i in range(0, 104):

weeks\_cost.append(add\_container[i]\*200+add\_arm[i]\*100+weekly\_maintenance\_arm[i]\*5+weekly\_maintenance\_container[i]\*10+v[i]\*10)

# 绘制表格

wb = Workbook()

sheet = wb.active

sheet.title='问题三：1-104周的结果数据'

sheet.append(['周数', '购买的容器艇数量','现有的操作手数量', '购买的操作手数量','保养的操作手数量','保养的容器艇数量','参与训练的操作手数量','总成本（单位：元）'])

for i in range(0,104):

sheet.append([weeks[i],add\_container[i],existing\_arms[i],add\_arm[i],weekly\_maintenance\_arm[i],weekly\_maintenance\_container[i],v[i],weeks\_cost[i]])

sheet.append(["1-104周总和",sum(add\_container),sum(existing\_arms),sum(add\_arm),sum(weekly\_maintenance\_arm),sum(weekly\_maintenance\_container),sum(v),sum(weeks\_cost)])

wb.save('问题三：1-104周的结果数据.xlsx')

wb = load\_workbook('问题三：1-104周的结果数据.xlsx')

sheet = wb.active

#表格列宽自适应

lks = []

for i in range(1, sheet.max\_column + 1):

lk = 1

for j in range(1, sheet.max\_row + 1):

sz = sheet.cell(row=j, column=i).value

if isinstance(sz, str):

lk1 = len(sz.encode('gbk'))

else:

lk1 = len(str(sz))

if lk < lk1:

lk = lk1

# print(lk)

lks.append(lk)

# 第二步：设置列宽

for i in range(1, sheet.max\_column + 1):

k = get\_column\_letter(i)

sheet.column\_dimensions[k].width = lks[i - 1] + 2

wb.close()

wb.save('问题三：1-104周的结果数据.xlsx')

**附录五：问题三代码生成的表格**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 周数 | 购买的容器艇数量 | 现有的操作手数量 | 购买的操作手数量 | 保养的操作手数量 | 保养的容器艇数量 | 参与训练的操作手数量 | 总成本（单位：元） |
| 第1周 | 0 | 50 | 14 | 5 | 2 | 15 | 1595 |
| 第2周 | 0 | 60 | 0 | 40 | 7 | 0 | 270 |
| 第3周 | 0 | 56 | 0 | 40 | 7 | 0 | 270 |
| 第4周 | 6 | 56 | 36 | 26 | 4 | 38 | 5350 |
| 第5周 | 0 | 88 | 0 | 24 | 0 | 0 | 120 |
| 第6周 | 0 | 80 | 0 | 56 | 8 | 0 | 360 |
| 第7周 | 0 | 76 | 0 | 56 | 8 | 0 | 360 |
| 第8周 | 2 | 72 | 8 | 43 | 5 | 9 | 1555 |
| 第9周 | 0 | 76 | 0 | 24 | 0 | 0 | 120 |
| 第10周 | 0 | 72 | 0 | 48 | 6 | 0 | 300 |
| 第11周 | 0 | 68 | 0 | 48 | 6 | 0 | 300 |
| 第12周 | 3 | 64 | 12 | 35 | 3 | 13 | 2135 |
| 第13周 | 0 | 72 | 0 | 24 | 0 | 0 | 120 |
| 第14周 | 0 | 68 | 0 | 48 | 6 | 0 | 300 |
| 第15周 | 0 | 64 | 0 | 48 | 6 | 0 | 300 |
| 第16周 | 0 | 64 | 0 | 40 | 4 | 0 | 240 |
| 第17周 | 0 | 60 | 0 | 24 | 0 | 0 | 120 |
| 第18周 | 0 | 56 | 0 | 36 | 3 | 0 | 210 |
| 第19周 | 5 | 52 | 12 | 31 | 2 | 13 | 2505 |
| 第20周 | 19 | 60 | 100 | 11 | 0 | 105 | 14905 |
| 第21周 | 0 | 156 | 44 | 37 | 0 | 47 | 5055 |
| 第22周 | 0 | 188 | 0 | 104 | 5 | 0 | 570 |
| 第23周 | 0 | 180 | 0 | 112 | 7 | 0 | 630 |
| 第24周 | 7 | 172 | 16 | 91 | 2 | 17 | 3645 |
| 第25周 | 0 | 180 | 0 | 72 | 0 | 0 | 360 |
| 第26周 | 0 | 168 | 0 | 116 | 11 | 0 | 690 |
| 第27周 | 0 | 164 | 0 | 128 | 14 | 0 | 780 |
| 第28周 | 0 | 160 | 0 | 120 | 12 | 0 | 720 |
| 第29周 | 0 | 156 | 0 | 92 | 5 | 0 | 510 |
| 第30周 | 0 | 148 | 0 | 124 | 13 | 0 | 750 |
| 第31周 | 0 | 144 | 0 | 124 | 13 | 0 | 750 |
| 第32周 | 0 | 140 | 0 | 112 | 10 | 0 | 660 |
| 第33周 | 0 | 136 | 0 | 92 | 5 | 0 | 510 |
| 第34周 | 0 | 132 | 0 | 112 | 10 | 0 | 660 |
| 第35周 | 0 | 128 | 0 | 108 | 9 | 0 | 630 |
| 第36周 | 0 | 124 | 0 | 100 | 7 | 0 | 570 |
| 第37周 | 0 | 120 | 0 | 72 | 0 | 0 | 360 |
| 第38周 | 0 | 116 | 0 | 88 | 4 | 0 | 480 |
| 第39周 | 1 | 112 | 0 | 84 | 3 | 0 | 650 |
| 第40周 | 6 | 108 | 0 | 68 | 0 | 0 | 1540 |
| 第41周 | 0 | 104 | 0 | 44 | 0 | 0 | 220 |
| 第42周 | 0 | 96 | 0 | 56 | 3 | 0 | 310 |
| 第43周 | 0 | 92 | 0 | 56 | 3 | 0 | 310 |
| 第44周 | 5 | 88 | 16 | 43 | 0 | 17 | 2985 |
| 第45周 | 0 | 100 | 0 | 40 | 0 | 0 | 200 |
| 第46周 | 0 | 92 | 0 | 52 | 3 | 0 | 290 |
| 第47周 | 5 | 88 | 16 | 47 | 2 | 17 | 3025 |
| 第48周 | 12 | 100 | 68 | 32 | 0 | 72 | 10080 |
| 第49周 | 0 | 160 | 28 | 54 | 0 | 30 | 3370 |
| 第50周 | 2 | 176 | 0 | 92 | 2 | 0 | 880 |
| 第51周 | 15 | 168 | 68 | 72 | 0 | 72 | 10880 |
| 第52周 | 18 | 228 | 116 | 78 | 0 | 122 | 16810 |
| 第53周 | 0 | 328 | 52 | 125 | 0 | 55 | 6375 |
| 第54周 | 5 | 360 | 0 | 180 | 0 | 0 | 1900 |
| 第55周 | 9 | 340 | 36 | 158 | 0 | 38 | 6570 |
| 第56周 | 13 | 356 | 68 | 156 | 0 | 72 | 10900 |
| 第57周 | 0 | 404 | 0 | 176 | 0 | 0 | 880 |
| 第58周 | 0 | 380 | 0 | 208 | 8 | 0 | 1120 |
| 第59周 | 1 | 364 | 0 | 204 | 7 | 0 | 1290 |
| 第60周 | 12 | 348 | 36 | 170 | 0 | 38 | 7230 |
| 第61周 | 0 | 368 | 12 | 159 | 0 | 13 | 2125 |
| 第62周 | 0 | 360 | 0 | 188 | 4 | 0 | 980 |
| 第63周 | 6 | 344 | 4 | 175 | 1 | 5 | 2535 |
| 第64周 | 12 | 332 | 56 | 149 | 0 | 59 | 9335 |
| 第65周 | 0 | 368 | 4 | 159 | 0 | 5 | 1245 |
| 第66周 | 0 | 352 | 0 | 188 | 6 | 0 | 1000 |
| 第67周 | 2 | 336 | 0 | 180 | 4 | 0 | 1340 |
| 第68周 | 11 | 320 | 36 | 154 | 0 | 38 | 6950 |
| 第69周 | 0 | 340 | 0 | 148 | 0 | 0 | 740 |
| 第70周 | 0 | 320 | 0 | 180 | 8 | 0 | 980 |
| 第71周 | 0 | 304 | 0 | 168 | 5 | 0 | 890 |
| 第72周 | 10 | 292 | 16 | 151 | 1 | 17 | 4535 |
| 第73周 | 0 | 292 | 12 | 123 | 0 | 13 | 1945 |
| 第74周 | 1 | 288 | 0 | 152 | 4 | 0 | 1000 |
| 第75周 | 11 | 276 | 40 | 130 | 0 | 42 | 7270 |
| 第76周 | 16 | 300 | 92 | 123 | 0 | 97 | 13985 |
| 第77周 | 0 | 376 | 36 | 154 | 0 | 38 | 4750 |
| 第78周 | 10 | 388 | 20 | 195 | 1 | 21 | 5195 |
| 第79周 | 16 | 388 | 88 | 167 | 0 | 93 | 13765 |
| 第80周 | 22 | 456 | 124 | 189 | 0 | 131 | 19055 |
| 第81周 | 0 | 552 | 48 | 229 | 0 | 51 | 6455 |
| 第82周 | 9 | 568 | 8 | 287 | 2 | 9 | 4145 |
| 第83周 | 18 | 548 | 88 | 247 | 0 | 93 | 14565 |
| 第84周 | 25 | 608 | 136 | 261 | 0 | 143 | 21335 |
| 第85周 | 0 | 708 | 52 | 301 | 0 | 55 | 7255 |
| 第86周 | 6 | 720 | 0 | 364 | 2 | 0 | 3040 |
| 第87周 | 11 | 684 | 28 | 330 | 0 | 30 | 6950 |
| 第88周 | 19 | 676 | 84 | 311 | 0 | 89 | 14645 |
| 第89周 | 0 | 724 | 24 | 322 | 0 | 26 | 4270 |
| 第90周 | 7 | 708 | 0 | 360 | 3 | 0 | 3230 |
| 第91周 | 10 | 672 | 36 | 318 | 0 | 38 | 7570 |
| 第92周 | 24 | 672 | 100 | 311 | 0 | 105 | 17405 |
| 第93周 | 0 | 736 | 36 | 318 | 0 | 38 | 5570 |
| 第94周 | 4 | 732 | 0 | 376 | 5 | 0 | 2730 |
| 第95周 | 10 | 696 | 20 | 339 | 0 | 21 | 5905 |
| 第96周 | 25 | 680 | 104 | 314 | 0 | 110 | 18070 |
| 第97周 | 1 | 748 | 60 | 321 | 0 | 63 | 8435 |
| 第98周 | 8 | 764 | 0 | 380 | 0 | 0 | 3500 |
| 第99周 | 14 | 724 | 48 | 345 | 0 | 51 | 9835 |
| 第100周 | 20 | 736 | 96 | 335 | 0 | 101 | 16285 |
| 第101周 | 1 | 792 | 40 | 354 | 0 | 42 | 6390 |
| 第102周 | 7 | 788 | 0 | 392 | 0 | 0 | 3360 |
| 第103周 | 16 | 748 | 44 | 361 | 0 | 47 | 9875 |
| 第104周 | 0 | 752 | 0 | 344 | 0 | 0 | 1720 |