

基于优化模型与数据分析的农作物种植策略研究

——以华北山区某乡村为例

摘 要

立足于华北山区某乡村低温和耕地资源有限的实际条件，本文采用线性规划、随机优化、系统聚类分析和相关分析的方法，在考虑各种干扰因素的情况下，探索了 2024 至 2030 年农作物种植和优化策略，以促进农业经济的可持续发展，并提出了相应的优化方案。

针对问题一，针对产品滞销或降价销售的情况，本文考虑作物连续种植减产风险和豆类作物对土壤的增益效应，构建了**线性规划模型**，为乡村提供了**优化种植策略**。模型的结果表明，该模型能够提升农作物种植的总收益。

针对问题二，受市场与气候的影响，考虑了销售量、成本波动及作物产量的不确定性，本文构建了**随机优化模型**，设计出适应未来变化的种植策略。结果表明，该模型能够在不确定的市场环境中实现风险与收益的平衡，增强了乡村经济的抗风险能力。

针对问题三，针对作物间的替代与互补性，以及销售价格与成本的相关性，本文构建了**系统聚类分析和相关性分析模型**，为乡村提供了定制化的种植策略。结果发现，该模型给出的优化种植策略可以实现经济效益和实现生态效益的双重提升。

本文的研究亮点在于综合利用了**线性规划、随机优化、聚类分析和相关分析**等模型，给出了不同影响因素下农作物的种植策略，直接促进了**农业的经济效益、环境保护与资源合理利用**。未来还可以在此基础上，进一步研究受人力资源限制和自然糟害风险伴随下的农作物种植策略问题，以保障我国农业的可持续发展。

关键词：农作物；线性规划；随机优化；系统聚类；相关分析；种植策略

一、问题重述

1.1 问题背景

位于华北山区的某乡村，面对温度偏低、耕地资源有限的环境，致力于通过有机种植优化产业结构，推动经济可持续发展。乡村拥有多样化的耕地类型，总面积 1201 亩，分布于 34 个地块，包括平旱地、梯田、山坡地和水浇地，以及 20 个大棚。耕地类型和作物生长规律限制了种植周期和作物选择，但同时也提供了通过科学规划提升产量和效益的机遇。

1.2 问题重述

问题 1 需求稳定下，探索 2024 至 2030 年最优种植策略，面对产品滞销或降价销售两种情景，需考虑作物连续种植的减产风险和豆类作物对土壤的增益效应。

问题 2 考虑市场与气候变动，预测小麦、玉米销售量增长，其他作物销量和成本的波动，及各类作物产量的不确定性，设计适应未来变化的种植策略，平衡风险与收益。

问题 3 在问题 2 的基础上，进一步分析作物间的替代与互补性，销售价格与成本的相关性，通过模拟求解，优化种植策略，实现经济效益与生态效益的双重提升。

此研究旨在通过数学模型，结合市场预测与生态农业原则，为乡村提供定制化的种植策略，促进经济持续稳定增长，同时兼顾环境保护与资源合理利用。

二、问题分析

2.1 问题一的分析

针对问题 1，我们深入分析了乡村的耕地资源分布、作物种植周期限制以及连续种植减产的风险。在滞销或降价销售两种不同情景下，通过构建线性规划模型，以作物种植面积和类型为决策变量，目标是在满足耕地和作物种植周期约束的条件下，实现总收益最大化。模型中充分考虑了作物种植的生态约束，如豆类作物对土地的增益效果，确保作物种植策略既科学又经济。

2.2 问题二的分析

针对问题 2，面临一个更为动态和不确定的市场与气候环境。农作物销售量、种植成本和销售价格的波动性，以及未来市场趋势的不确定性，构成了模型构建的复杂背景。采用随机优化模型，我们引入了销售量、成本、产量的随机性，通过模拟不同情景下的作物种植策略，评估其经济效益与风险，目标是在未来市场波动中，优化资源配置，实现长期收益最大化。模型不仅考虑了作物种植的经济性，也兼顾了对市场变化的适应性，为乡村的可持续发展提供科学依据。

2.3 问题三的分析

针对问题 3，通过聚类分析深入作物市场网络，识别销售量、价格与成本间的内在联系，考量作物的可替代性与互补性。在问题 2 基础上，模拟市场动态与作物间交互，评估种植策略的经济与生态影响，旨在优化资源配置，实现市场适应与生态平衡的双重目标。聚类分析揭示关键驱动因素，助力乡村精准决策，促进可持续发展。

三、模型假设

- ❖ 轮作周期确定性：所有地块三年内轮作至少一次豆类作物。
- ❖ 作物替代简化：仅按大类考虑作物替代，简化具体品种分析。
- ❖ 种植成本线性增长：成本每年增长 5%，不考虑非线性经济波动。
- ❖ 参数正态分布：销售量等参数遵循正态分布，均值标准差按题目设定。
- ❖ 销售量价格稳定性：除特殊作物，其他作物销量与价格保持 2023 年水平。
- ❖ 气候影响平均化：作物产量 $\pm 10\%$ 变化仅因常规气候，排除极端天气影响。

四、符号说明

符号	说明	单位
T	时间段（即 2024 年至 2030 年）	
L	地块集合（平旱地、梯田、山坡地、水浇地以及大棚）	
C	作物集合，包括粮食类作物、蔬菜类作物和食用菌类作物	
z_{lct}	作物 c 在时间段 t 在地块 l 上种植的决策变量	
x_{lct}	作物 c 在时间段 t 在地块 l 上种植的数量	亩
y_{ct}	作物 c 在时间段 t 的总种植面积	亩
p_c	作物 c 的销售价格	元/斤
q_c	作物 c 的预期销售量	斤
α_c	作物 c 的亩产量	斤/亩
c_c	作物 c 的种植成本	元/亩

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 数据的清理与整理

为了保证数据的质量，我们对收集到的信息进行了详尽的验证和清理工作。主要步骤如下：

数据去重：检查并移除了重复的记录，确保每条数据的唯一性。

缺失值处理：对于缺失的关键字段，如作物种植成本或亩产量，我们参考了类似作物的历史数据进行了合理的填充。

异常值检测：通过对比历史数据，识别并修正了明显偏离正常范围的数值。

为了便于后续的数学模型构建及求解，我们将所有有效数据进行了系统化整理：

耕地类型分类：依据耕地的不同类型（平旱地、梯田、山坡地、水浇地）和特征，整理出了每种类型耕地的具体面积和数量。

大棚种植信息汇总：对于 16 个普通大棚和 4 个智慧大棚，分别列出了它们各自适合种植的作物种类及其种植季节。

作物经济指标整理：整理了 2023 年每种作物的种植成本、亩产量、销售价格以及预期销售量，并将其按地块类型分类汇总，以便于后续模型构建时使用。

豆类作物种植记录：特别整理了 2023 年所有地块上豆类作物的种植情况，确保在未来七年的种植计划中，每块地能在三年内至少种植一次豆类作物，以此满足生态需求。

通过上述数据的清理与整理的内容，删除了重复的记录、原始数据不存在缺失值得到了如下数据表。

表 1 整理后的数据表

作物编号	作物名称	地块类型	种植季次	亩产量/斤	种植成本/(元/亩)	销售单价/(元/斤)
1	黄豆	平旱地	单季	400	400	2.50-4.00
2	黑豆	平旱地	单季	500	400	6.50-8.50
3	红豆	平旱地	单季	400	350	7.50-9.00
4	绿豆	平旱地	单季	350	350	6.00-8.00
5	爬豆	平旱地	单季	415	350	6.00-7.50
6	小麦	平旱地	单季	800	450	3.00-4.00
7	玉米	平旱地	单季	1000	500	2.50-3.50
8	谷子	平旱地	单季	400	360	6.00-7.50
9	高粱	平旱地	单季	630	400	5.50-6.50
10	黍子	平旱地	单季	525	360	6.50-8.50
11	荞麦	平旱地	单季	110	350	30.00-50.00
12	南瓜	平旱地	单季	3000	1000	1.00-2.00
13	红薯	平旱地	单季	2200	2000	2.50-4.00
14	莜麦	平旱地	单季	420	400	5.00-6.00
15	大麦	平旱地	单季	525	350	3.00-4.00
16	水稻	水浇地	单季	500	680	6.00-8.00
17	豇豆	水浇地	第一季	3000	2000	7.00-9.00
18	刀豆	水浇地	第一季	2000	1000	5.50-8.00
19	芸豆	水浇地	第一季	3000	2000	5.00-8.00
20	土豆	水浇地	第一季	2000	2000	3.00-4.50
21	西红柿	水浇地	第一季	2400	2000	5.00-7.50
22	茄子	水浇地	第一季	6400	2000	5.00-6.00
23	菠菜	水浇地	第二季	2700	2300	4.80-6.70
24	青椒	水浇地	第一季	2400	1600	4.00-6.50
25	菜花	水浇地	第一季	3300	2400	5.00-6.00
26	包菜	水浇地	第一季	3700	2900	5.50-7.50
27	油麦菜	水浇地	第一季	4100	1600	4.00-6.00
28	小青菜	水浇地	第一季	3200	1600	5.00-6.50
29	黄瓜	水浇地	第一季	12000	2900	6.00-8.00
30	生菜	水浇地	第一季	4100	1600	4.50-6.00
31	辣椒	水浇地	第一季	1600	1000	6.00-8.50
32	空心菜	水浇地	第一季	10000	4100	3.00-6.00
33	黄心菜	水浇地	第一季	5000	2000	4.00-5.00
34	芹菜	水浇地	第二季	5500	900	3.20-4.80
35	大白菜	水浇地	第二季	5000	2000	2.00-3.00
36	白萝卜	水浇地	第二季	4000	500	2.00-3.00

作物编号	作物名称	地块类型	种植季次	亩产量/斤	种植成本/(元/亩)	销售单价/(元/斤)
37	红萝卜	水浇地	第二季	3000	500	2.50-4.00
38	榆黄菇	普通大棚	第二季	5000	3000	50.00-65.00
39	香菇	普通大棚	第二季	4000	2000	18.00-20.00
40	白灵菇	普通大棚	第二季	10000	10000	14.00-18.00
41	羊肚菌	普通大棚	第二季	1000	10000	80.00-120.00

为了直观展示不同作物的经济效益，绘制了包含亩产量、种植成本以及销售单价这三个关键经济指标的点线图。下图显示了各种作物在这三个维度上的表现情况，为我们进一步分析作物种植的经济效益提供了可视化支持。

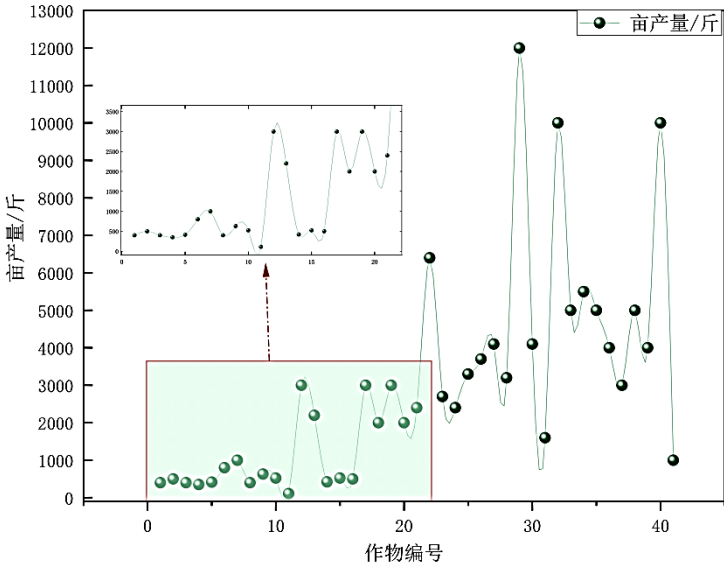


图 1 亩产量的经济效益关系图

此图展示了不同作物编号对应的亩产量，可以看出诸如黄豆、黑豆等豆类作物在平旱地上的平均亩产量较为稳定，而像小麦、玉米这样的粮食作物则具有较高的亩产量。

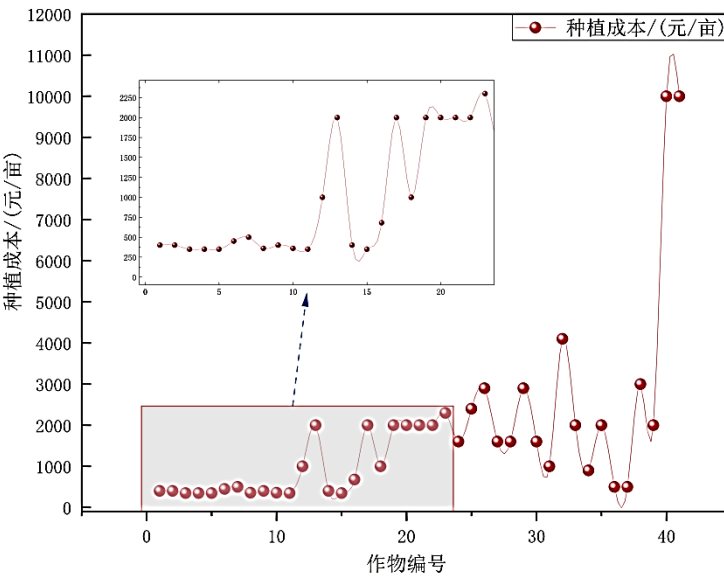


图 2 作物种植成本分析图

这张图表反映了各作物种植成本的差异，可以看到，虽然多数粮食作物的种植成

本相似，但是像荞麦这类特殊作物的成本却显著高于其它作物。这表明在选择种植作物时需要考虑其经济效益。

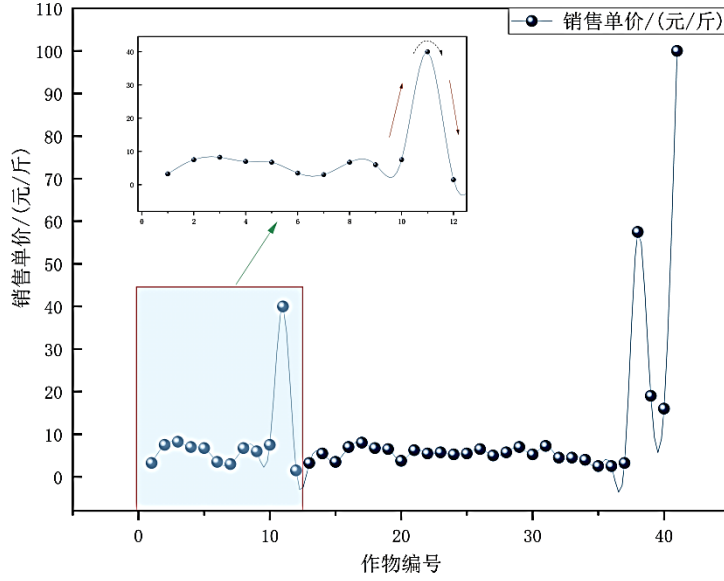


图 3 作物销售单价趋势图

通过本图可以发现，作物的销售单价存在着较大的差异。例如，黑豆和红豆等豆类作物因其营养价值较高，销售单价也相对较高，而南瓜等作物虽然产量大，但销售单价却处于较低水平。这有助于规划作物种植以最大化收益。

经过数据的清理与整理，获得了清晰、准确的数据集，为后续的分析 and 建模奠定了坚实的基础。为了进一步挖掘数据背后的潜在价值，并为该乡村提供科学合理的农作物种植策略，我们在此基础上建立了数学模型。该模型旨在通过优化种植方案来最大化总收益，同时考虑到了地块类型、作物种类、种植成本、预期销售量及销售价格等多个因素的影响。接下来，我们将详细介绍模型的建立过程及其求解方法。

5.1.2 模型的建立

Step 1. 确定目标函数

模型的目标是最大化收益，收益计算公式如下

$$\text{Maximize } Z = \sum_{t \in T} \sum_{c \in C} \sum_{l \in L} (p_c \cdot \alpha_c \cdot x_{lct} - c_c \cdot x_{lct}) \quad (1)$$

其中， $p_c \cdot \alpha_c \cdot x_{lct}$ 表示作物 c 在地块 l 和时间段 t 的销售收入，而 $c_c \cdot x_{lct}$ 表示对应的种植成本。

Step 2. 确定约束条件

(1) 地块面积约束

$$\sum_{c \in C} x_{lct} \leq \text{Area}_l, \forall l \in L, \forall t \in T \quad (2)$$

每个地块的总种植面积不能超过其实际面积。

(2) 作物销售量约束

$$\sum_{l \in L} y_{ct} = \sum_{l \in L} (\alpha_c \cdot x_{lct}) \leq q_c, \forall c \in C, \forall t \in T \quad (3)$$

每种作物的总产量不能超过其预期销售量。

(3) 豆类作物种植约束

$$\sum_{l \in L} x_{lct} \geq 1, \forall t \in T, \forall c \in Legumes, \text{if } t \bmod 3 = 0 \quad (4)$$

每块地每三年至少种植一次豆类作物。

(4) 重茬种植约束

$$x_{lct} \cdot x_{lc(t-1)} = 0, \forall l \in L, \forall c \in C, \forall t \in T \quad (5)$$

作物在同一地块不能连续重茬。

(5) 最小种植面积约束

$$x_{lct} \geq MinArea, \forall l \in L, \forall c \in C, \forall t \in T \quad (6)$$

每种作物在单个地块的种植面积不宜过小。

(6) 滞销处理约束

$$Revenue_{ct} = q_c \cdot p_c + (\alpha_c \cdot x_{lct} - q_c) \cdot \frac{p_c}{2}, \text{if } \alpha_c \cdot x_{lct} > q_c \quad (7)$$

如果作物产量超过预期销售量，则超过部分按半价销售。

通过上述模型建立的步骤结合解决该问题的算法逻辑，可以编写出代码，完成这个优化问题，该代码的流程图如下：

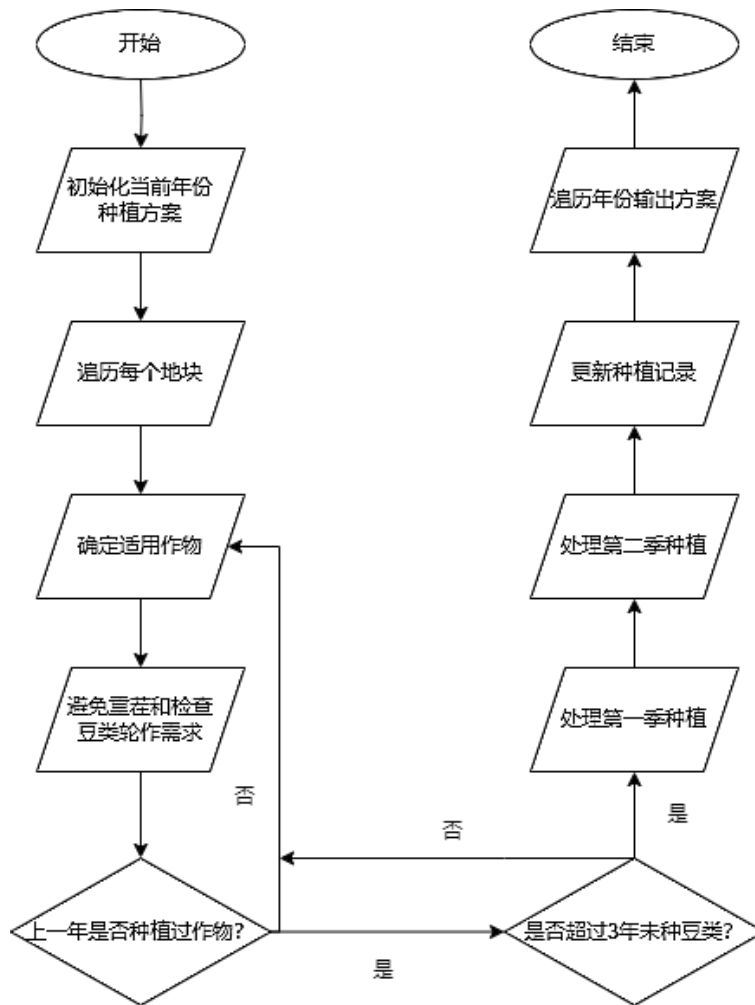


图 4 种植面积优化算法_1 流程图

5. 1. 3 模型的求解

为了求解上述建立的优化模型，设计一个算法来迭代地评估每个地块在给定年份中的最佳作物选择。

算法首先初始化所有必要的数据结构，包括记录每块地历年种植情况的历史数据，以及未来几年的种植计划。

随后，算法通过遍历每个年份和每个地块，根据地块特性筛选出适用的作物列表，并排除前一年种植过的作物以避免连作。

如果地块已经有三年未种植豆科作物，则优先考虑种植豆科作物。

接着，算法会根据作物的预期收益和最小种植面积要求来决定每种作物的具体种植面积。

最后，如果在某个季节未能找到合适的作物，则随机选择一种适用作物进行种植。

以下是植面积优化算法的伪代码：

表 2 种植面积优化核心算法_1 伪代码

伪代码：种植面积优化核心算法 1
<pre># 初始化变量和数据结构 years = range(2024, 2031) # 年份范围 # 初始化地块和作物数据 Initialize_plot_and_crop_data() # 定义豆科作物集合 Define_legume_crops_set() # 初始化预期销售因子和最小地块面积 Initialize_expected_sales_factor() Initialize_min_plot_area() # 初始化上一次种植的作物信息 Initialize_last_crop_planted() Initialize_last_legume_year() # 循环遍历年份 for year in years: # 初始化年度两季的种植计划 Initialize_year_plan(year) # 为每年的两个季节初始化计划 for plot in plots: # 循环每个地块 # 获取适用的作物和检查作物轮作规则 Get_applicable_crops_and_check_rotation_rules(plot) # 处理第一季种植 Handle_first_season_planting(plot, year) # 基于收入找到最佳作物种植 Find_best_crop_to_plant(plot, year) # 更新种植面积和上一次种植的作物 Update_planting_areas_and_last_crop_planted(plot, year) # 处理第二季种植，逻辑与第一季类似 Handle_second_season_planting(plot, year)</pre>

伪代码：种植面积优化核心算法_1

保存年度计划

Save_yearly_plan(year)

定义独特的作物名称和行目标

Define_unique_crop_names_and_row_targets()

再次循环遍历年份来处理和保存计划

for year in years:

 # 获取年度计划

 Get_year_plan(year)

 # 合并相同作物的种植面积

 Combine_same_crops_planting_areas()

 # 限制输出列数

 Limit_number_of_columns_in_output()

 # 汇总并按比例缩放行以满足目标

 Sum_and_scale_rows_to_meet_target()

 # 保存组合计划到Excel 文件，分别为第一季和第二季

 Save_combined_plans_to_excel(year, 'first_season')

 Save_combined_plans_to_excel(year, 'second_season')

(1) 局部分析

通过运行代码，得到了各作物在各种种植地块的种植面积，以 2024 年各种种植地块的种植面积数据为例，进行简要分析。

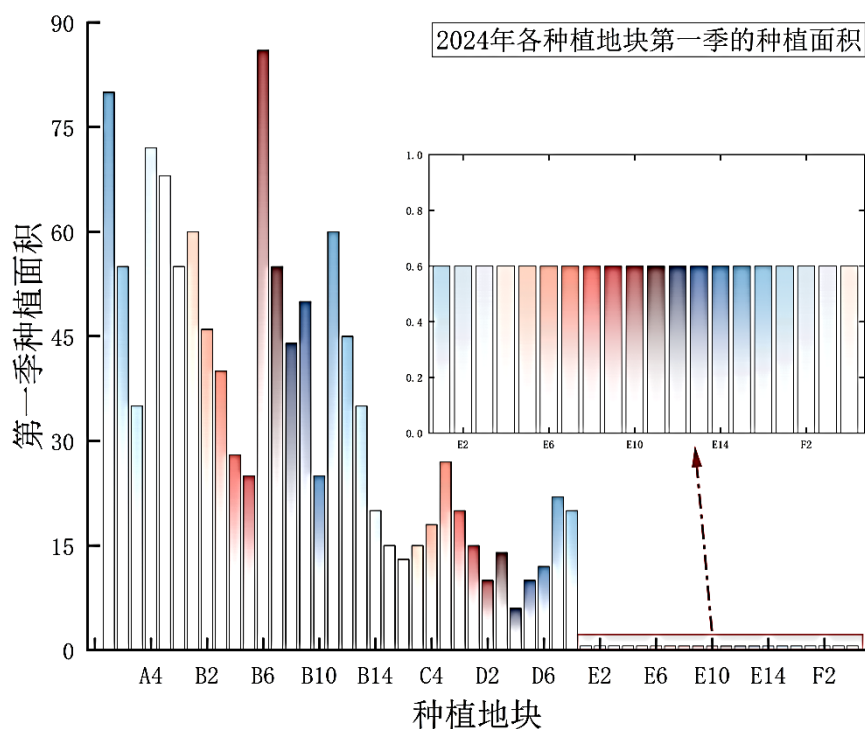


图 5 2024 年第一季各地块各作物的种植面积柱状图

该柱状图显示了 2024 年第一季各地块的种植面积分布情况。大型地块(A1 至 D8)的种植面积从 6 亩到 86 亩不等，反映了不同地块种植规模的差异。

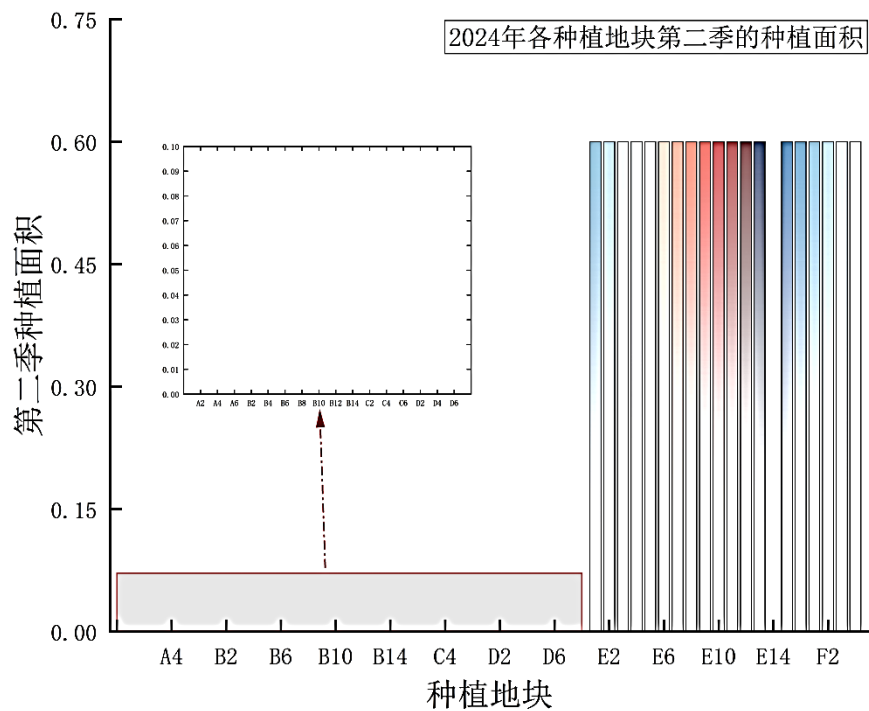
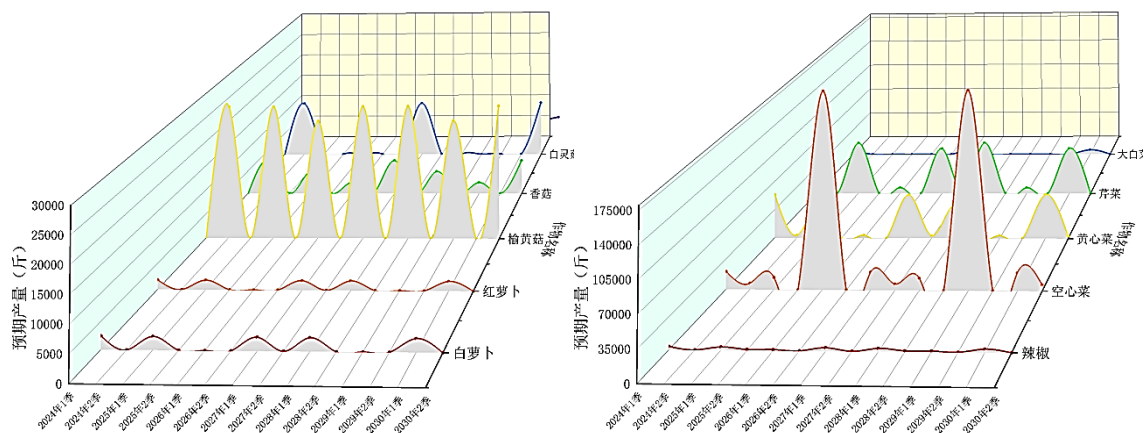


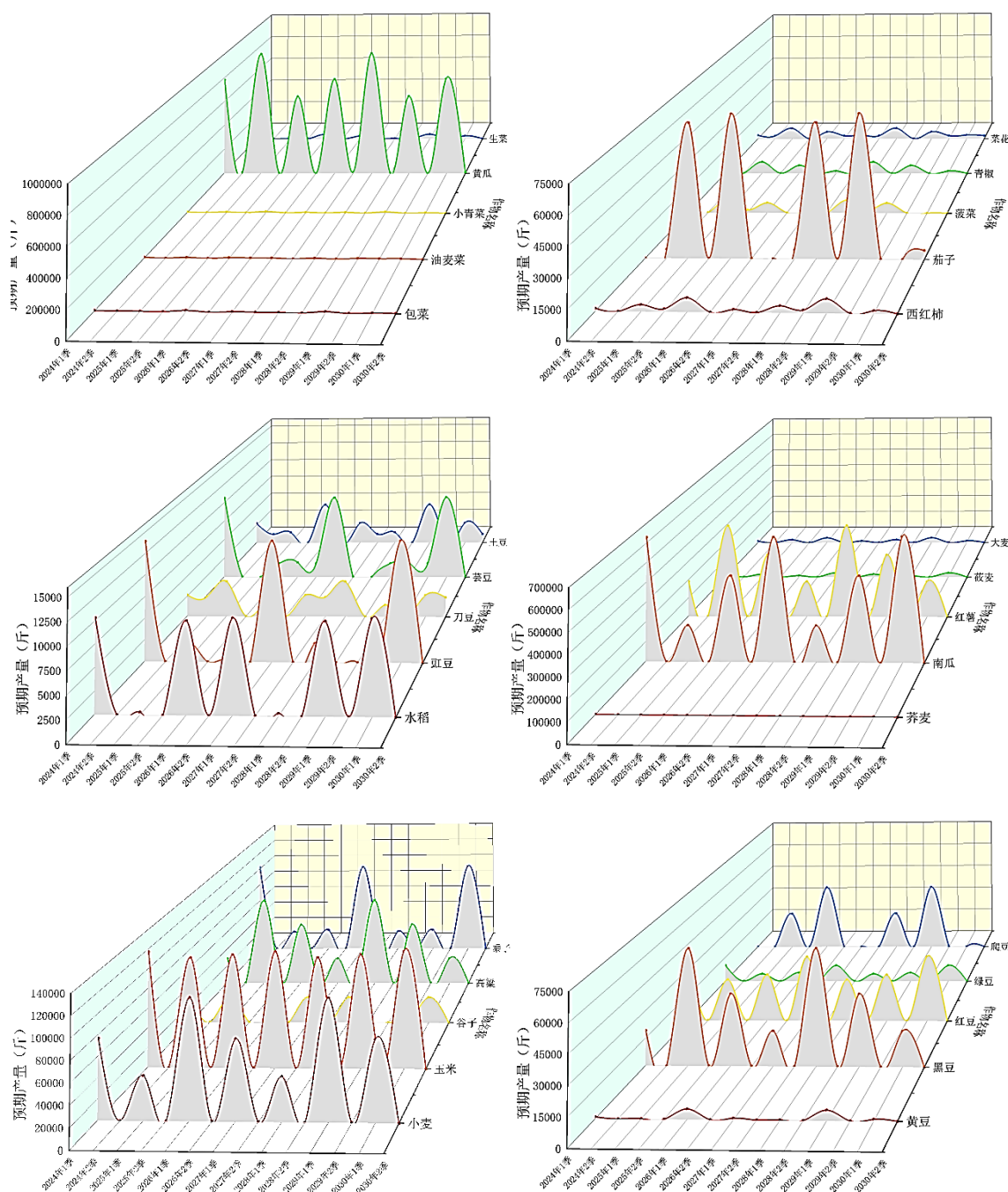
图 6 2024 年第二季各地块各作物的种植面积柱状图

该柱状图显示了 2024 年第二季各地块的种植面积分布情况。小型地块(E1 至 F4)的种植面积均为 0.6 亩，除了 E14 地块未种植任何作物。

(2) 全局分析

为了更直观地展示这些地块在不同季度间的种植面积变化，使用三维瀑布图来帮助分析。三维瀑布图不仅能够清晰地显示单个地块在相邻季度之间的种植面积变化，还能让我们从整体上把握所有地块种植面积动态趋势。通过这种方式，我们可以更容易地识别出哪些地块的种植模式发生了显著改变，以及这些改变是否具有规律性或特定目的。





首先，我们可以看出，小麦、玉米、高粱、谷子、黍子、红豆、南瓜、红薯、莜麦、大麦、水稻、刀豆、芸豆、土豆、西红柿、茄子、菠菜、青椒、菜花、包菜、油麦菜、小青菜、黄瓜、生菜、辣椒、空心菜、黄心菜、芹菜、大白菜、白萝卜、红萝卜、榆黄菇、香菇、白灵菇、羊肚菌等作物在不同年份和季度的产量存在显著差异。其中，小麦、玉米、高粱、谷子、红豆、南瓜、红薯、莜麦、大麦、水稻、刀豆、芸豆、土豆、西红柿、茄子、菠菜、青椒、菜花、包菜、油麦菜、小青菜、黄瓜、生菜、辣椒、空心菜、黄心菜、芹菜、大白菜、白萝卜、红萝卜、榆黄菇、香菇、白灵菇、羊肚菌等作物在 2024 年至 2030 年的产量总体上呈现上升趋势，而其余作物则有不同程度的下降。

其次，我们注意到，虽然大部分作物在两个季度之间存在一定的产量波动，但是总体来看，它们的产量相对稳定。例如，黄豆、黑豆、红豆、绿豆、爬豆、小麦、玉米、谷子、高粱、黍子、荞麦、南瓜、红薯、莜麦、大麦、水稻、豇豆、刀豆、芸豆、土豆、西红柿、茄子、菠菜、青椒、菜花、包菜、油麦菜、小青菜、黄瓜、生菜、辣椒、空心菜、黄心菜、芹菜、大白菜、白萝卜、红萝卜、榆黄菇、香菇、白灵菇、羊肚菌等作物在 2024 年第一季到 2030 年第二季期间的产量基本保持不变。然而，也有一些例外情况，如 2025 年第二季至 2026 年第一季间，部分作物出现了大幅度增长或减少的情况。

最后，通过对这些数据的分析，可以得出以下结论：

乡村管理者应关注不同作物的生长周期和产量波动，以便合理安排种植计划；

针对产量较低的作物，应采取措施提高其产量，如改善耕作技术、施肥等；

对于产量较高的作物，可考虑扩大种植规模，以获得更高的经济效益；

在选择种植品种时，应充分考虑市场供需状况及价格变动等因素，确保收益最大化。

5.2 问题二模型的建立与求解

5.2.1 模型的建立

Step 1. 确定目标函数

模型的目标是最大化收益，收益计算公式如下

$$\text{Maximize } R = \sum_{t \in T} \sum_{c \in C} \sum_{l \in L} (q_{ct} \cdot p_c - c_c \cdot A_l) \cdot z_{lct} \quad (8)$$

q_{ct} 是作物 c 在时间段 t 的预期销售量（总产量的 90%）； p_c 是作物 c 的销售价格；

c_c 是作物 c 的种植成本； A_l 是地块 l 的面积； z_{lct} 是作物 c 在时间段 t 在地块 l 上种植的决策变量（1 表示种植，0 表示不种植）。

Step 2. 确定约束条件

在问题 1 的约束条件基础上，再添加了如下约束条件：

（1）地块适用性

$$z_{lct} = 0, \forall l \in L, \forall t \in T \text{ if } t_l \neq T_c \quad (9)$$

只有符合地块类型要求的作物才能被种植。即，如果地块类型 t_l 与作物 c 的适用地块类型 T_c 不匹配，则不允许种植。

（2）季节性限制

$$z_{lct} = 0, \forall l \in L, \forall t \in T \text{ if } \text{season}(c) \neq t \quad (10)$$

某些作物只能在特定的季节种植。如果作物 c 只能在第一季或第二季种植，则必须遵守这一规定。

（3）最小种植面积

$$\sum_{c \in C} z_{lct} \cdot A_l > A_{\min}, \forall l \in L, \forall t \in T \quad (11)$$

每个作物在单个地块上的最小种植面积应满足一定的阈值，以确保经济效益。

(4) 轮作规则

$$\sum_{t'=t-3}^t z_{lct} \leq 1, \forall l \in L, c \in Legumes, t \in T \quad (12)$$

豆类作物在同一地块上的种植间隔至少为三年。

(5) 避免连作

$$z_{lct} \cdot z_{lc(t+1)} = 0, \forall l \in L, c \in C, t \in T \quad (13)$$

同一地块不得连续两年种植相同的作物。

(6) 地块面积限制

$$\sum_{c \in C} z_{lct} \cdot A_l \leq A_{\min}, \forall l \in L, \forall t \in T \quad (14)$$

地块上种植的所有作物总面积不能超过地块的实际面积。

在构建了以最大化收益为目标函数，并结合地块适用性、季节性限制、最小种植面积、轮作规则、避免连作以及地块面积限制等多种约束条件的模型之后，我们进一步设计了一个种植面积优化算法。

此算法通过年份循环的方式，确保每年内每一块地块都能依据上述约束条件得到最优化的作物种植安排。以下流程图展示了该算法的工作流程：

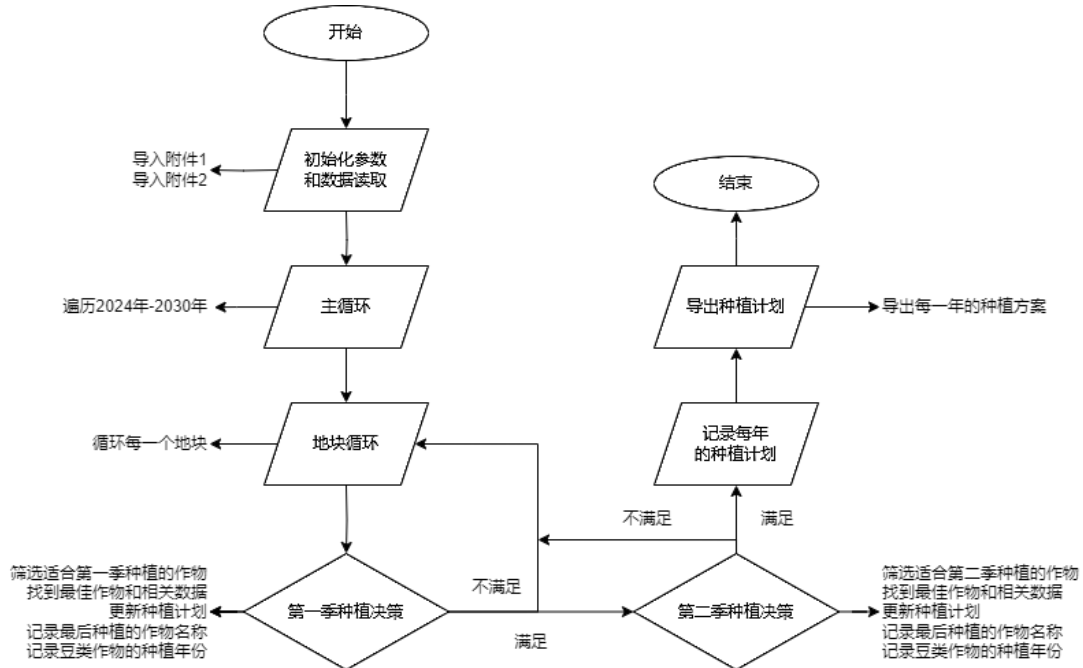


图 8 种植面积优化算法_2 流程图

5.2.2 模型的求解

为了实现上述模型的目标函数，并考虑到所有约束条件，我们设计了一套基于年份循环的种植面积优化算法。

该算法旨在每年度内针对每一块地块制定最优的作物种植计划，同时遵循地块适用性、季节性限制、最小种植面积、轮作规则、避免连作以及地块面积限制等约束。下面将通过伪代码形式展示这一算法的具体执行流程：

表 3 种植面积优化核心算法_2 伪代码

伪代码：种植面积优化核心算法_2

```

for each_year in range(2024, 2031): # 使用 range 函数生成年份列表
    for each_plot in plots: # 循环每一块地
        # 第一季种植决策
        # 筛选出适合第一季种植的作物
        first_season_crops = filter_crops_by_season('第一季',
plot_type)
        # 去除去年种植过的作物
        first_season_crops =
remove_last_year_crops(first_season_crops,
last_crop_planted[plot])
        # 如果距离上次种植豆科植物已超过 3 年
        if years_since_last_legume >= 3:
            # 只考虑豆科植物
            first_season_crops =
filter_to_legumes(first_season_crops)

        total_area_planted_first_season = 0 # 初始化已种植的总面积
为0
        for each_crop in first_season_crops: # 循环筛选出的每个作
物
            # 找到最佳作物和相关数据
            best_crop, _ = find_best_crop(crop, data, plot_area,
each_year, sales_factor)
            # 如果地块剩余面积大于已种植面积，且最佳作物的种植面积大于
0
            if plot_area > total_area_planted_first_season and
best_crop > 0:
                # 确定种植面积，不能超过剩余面积
                planting_area = min(remaining_area, plot_area)
                # 更新种植计划
                update_plan(first_season_plan[plot], best_crop,
planting_area)
                # 增加已种植的总面积
                total_area_planted_first_season +=
planting_area
                # 记录最后种植的作物名称
                last_crop_planted[plot] = crop_name(best_crop)
                # 如果作物是豆科植物，记录种植年份
                if crop is legume:
                    last_legume_year[plot] = each_year
            # 第二季种植决策（逻辑与第一季类似，代码省略）

```

伪代码：种植面积优化核心算法_2

```
# 记录每年的种植计划
yearly_plans[each_year].first_season = first_season_plan
yearly_plans[each_year].second_season =
second_season_plan
# 导出每年的种植计划
export_plans(yearly_plans[each_year], each_year)
```

通过运行代码，得到了各作物的最优种植方案，其中 2030 年各种种植地块的部分最优种植方案数据如下所示：

表 4 最优种植方案_部分

地块名	黄豆	黑豆	红豆	绿豆	爬豆	小麦	玉米	谷子	高粱	黍子
A1	0	0	0	0	0	80	0	0	0	0
A2	0	0	55	0	0	0	0	0	0	0
A3	0	0	0	0	0	35	0	0	0	0
A4	0	0	0	0	0	72	0	0	0	0
A5	0	0	0	0	0	0	68	0	0	0
A6	0	0	0	0	0	0	55	0	0	0
B1	0	54	0	0	0	0	0	0	0	0
B2	0	0	0	0	0	0	0	0	46	0
B3	0	0	0	0	0	0	0	40	0	0
B4	0	0	0	28	0	0	0	0	0	0
B5	0	0	0	0	0	0	0	0	0	25
B6	86	0	0	0	0	0	0	0	0	0
B7	0	0	0	55	0	0	0	0	0	0
B8	44	0	0	0	0	0	0	0	0	0
B9	0	0	0	0	0	0	0	50	0	0
B10	0	0	0	0	0	25	0	0	0	0

5.3 问题三模型的建立与求解

5.3.1 数据的清理与整理

在前两问的基础上，根据附件 1 中的“乡村种植的农作物”子表，对各作物的种植耕地类型进行归纳汇总（同一作物，适合种植在不同耕地，表明该作物的生存能力更强），定义一个新变量——作物的生存能力，得到下表：

作物编号	作物名称	生存能力
1	黄豆	3
2	黑豆	3
3	红豆	3
4	绿豆	3
5	爬豆	3
6	小麦	3
7	玉米	3
8	谷子	3

作物编号	作物名称	生存能力
9	高粱	3
10	黍子	3
11	荞麦	3
12	南瓜	3
13	红薯	3
14	莜麦	3
15	大麦	3
16	水稻	1
17	豇豆	4
18	刀豆	4
19	芸豆	4
20	土豆	4
21	西红柿	4
22	茄子	4
23	菠菜	4
24	青椒	4
25	菜花	4
26	包菜	4
27	油麦菜	4
28	小青菜	4
29	黄瓜	4
30	生菜	4
31	辣椒	4
32	空心菜	4
33	黄心菜	4
34	芹菜	4
35	大白菜	1
36	白萝卜	1
37	红萝卜	1
38	榆黄菇	1
39	香菇	1
40	白灵菇	1
41	羊肚菌	1

问题 3 指出，各种农作物之间可能存在一定的可替代性和互补性。

在原始数据中，对作物呈现固定影响程度的有：亩产量/斤、种植成本/(元/亩)、销售单价/(元/斤)，由于销售单价是一个固定的范围，所以我们取销售单价的平均数；再与前文定义的变量——作物的生存能力，构成了 41*6 大小的数据集（见附件）使用系统聚类，展现农作物直接的可替代性和互补性。

为了消除数据量纲带来的影响，使用 Z 得分标准化

Step 1. 计算均值

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

μ 是特征的平均值； x_i 是特征中的第 i 个数值； N 是特征中数值的总数。

Step 2. 计算标准差

计算特征的标准差，它是各数值与其均值之差的平方的平均值的平方根

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}}$$

σ 是特征的标准差； μ 是步骤 1 中计算得到的特征均值； x_i 是特征中的第 i 个数值； N 是特征中数值的总数。

Step 3. 应用 Z 得分公式

$$z_i = \frac{x_i - \mu}{\sigma}$$

问题 3 还指出预期销售量与销售价格、种植成本之间也存在一定的相关性，所以我们对预期销售量与销售价格、种植成本之间进行相关性分析。

5.3.2 模型的建立

(1) 系统聚类模型

① 欧氏距离

欧氏距离是两点在多维空间中的直线距离。它是通过计算两点间坐标的平方差之和再开平方得到的。

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

适用于测量在多维空间中两点之间的直线距离。通常用于数值型数据。

② 平方欧氏距离

平方欧氏距离是欧氏距离的平方形式，避免了开方运算，计算相对简单。

$$d_{SquaredEuclidean}(x, y) = \sum_{i=1}^n (x_i - y_i)^2$$

与欧氏距离类似，但在不需要精确距离值时使用，可以减少计算量。

③ 余弦相似度

余弦相似度用来衡量两个非零向量的角度，角度越小表示两个向量越相似。

$$\cosine(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

其中， $x \cdot y$ 表示向量点积， $\|x\|$ 、 $\|y\|$ 分别表示向量的范数；用于比较方向相似性，而非距离本身。适用于文本分析、推荐系统等领域。

④ 皮尔逊相关性

皮尔逊相关系数衡量两个变量线性相关程度，取值范围从 -1 到 1，1 表示完全正相关，-1 表示完全负相关，0 表示没有线性相关。

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

适用于评估变量间的线性关系，常用于数据分析和统计测试。

⑤切比雪夫距离

切比雪夫距离是指两点间最大的坐标轴上的绝对差值。

$$d_{Chebyshev}(x, y) = \max_i |x_i - y_i|$$

适用于当关心的是最大差异而不是总体差异的情况。

⑥块距离

块距离又称曼哈顿距离，是两点沿坐标轴方向移动的总距离。

$$d_{Manhattan}(x, y) = \sum_{i=1}^n |x_i - y_i|$$

适用于网格状布局或城市街区布局中的路径计算。

⑦明可夫斯基距离

明可夫斯基距离是欧氏距离和曼哈顿距离的推广，当 $p=1$ 时为曼哈顿距离，当 $p=2$ 时为欧氏距离。

$$d_{Manhattan}(x, y) = (\sum_{i=1}^n |x_i - y_i|^p)^{1/p}$$

适用于需要权衡不同距离度量时，通过改变 p 的值来调整距离度量。

(2) 相关分析模型

对于每一对农作物，计算它们之间的相关系数。最常用的相关系数是皮尔逊相关系数（Pearson correlation coefficient），它衡量了两个变量之间的线性关系强度。

皮尔逊相关系数衡量两个变量线性相关程度，取值范围从-1 到 1。

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

接近 1：表示强正相关，即当一种作物的需求上升时，另一种作物的需求也随之上升。

接近-1：表示强负相关，即当一种作物的需求上升时，另一种作物的需求下降。

接近 0：表示弱相关或无关。

基于计算出的相关系数，构建一个互补性矩阵 C ，其中 C_{ij} 表示作物 i 和作物 j 之间的互补程度。

5.3.3 模型的求解

(1) 系统聚类

利用 SPSS 中的“分类”模块，对上述不同的距离计算方式，进行系统聚类。得到如下聚类系谱图。

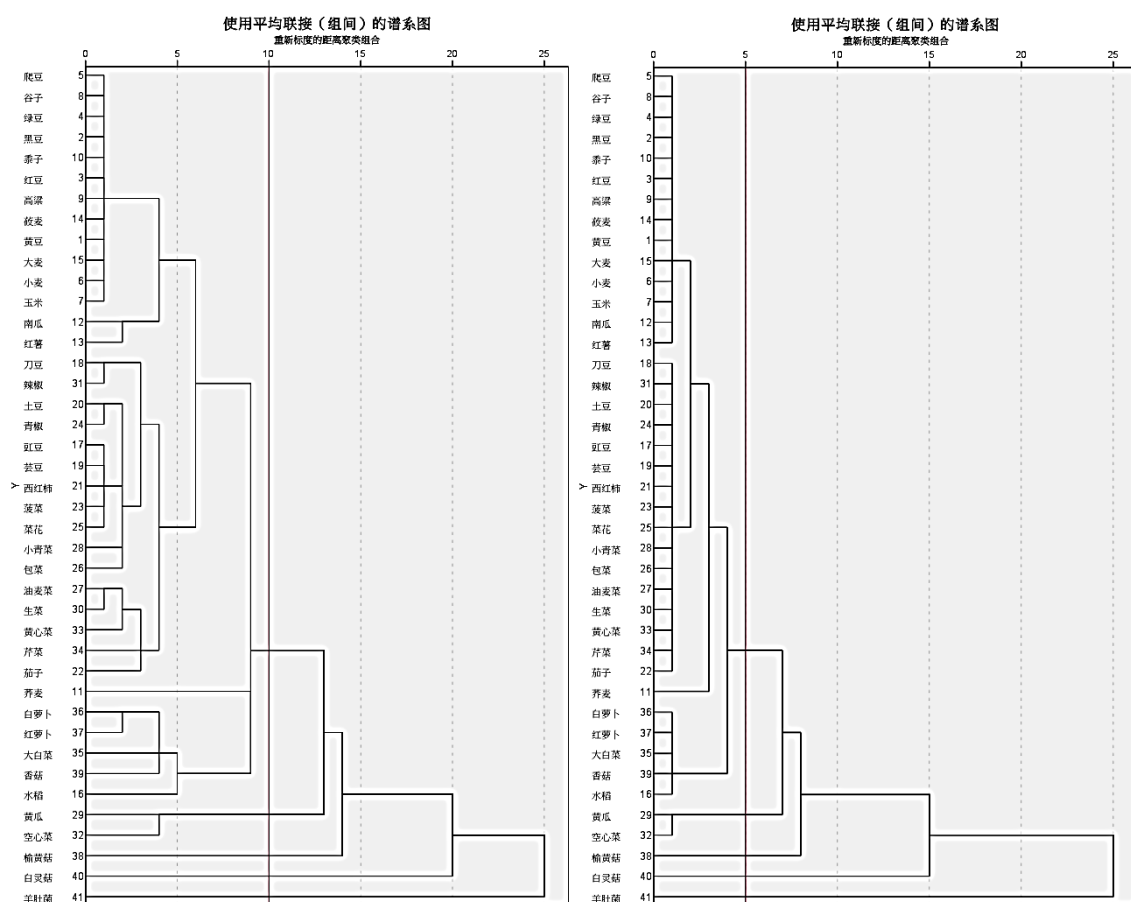


图 9 欧氏距离（左）和平方欧氏距离（右）下的分类情况

①欧氏距离

第一类作物：包含大多数主要粮食作物（如豆类、谷物）及一些蔬菜。这些作物具有较高的生存能力（3 或 4），适应性强，但平均销售单价较低。

第二类作物：黄瓜和空心菜。高产量（12000 斤/亩），销售单价不错，适合追求高产量和经济效益的种植者。

第三类作物：榆黄菇。亩产量适中（5000 斤/亩），销售单价高（57.5 元/斤），适合高端市场。

第四类作物：白灵菇。亩产量较高，但种植成本高，销售单价较高（16 元/斤）。

第五类作物：羊肚菌。亩产量低（1000 斤/亩），销售单价极高（100 元/斤），适合特定市场。

②平方欧氏距离

第一类作物：涵盖多种粮食作物和蔬菜。生存能力强，产量和种植成本各异，适合大规模种植。

第二类作物：黄瓜和空心菜。高产量（12000 斤/亩），适合追求高产量的种植者。

第三类作物：榆黄菇。高附加值，适合高端市场。

第四类作物：白灵菇。高附加值，适合高端市场。

第五类作物：羊肚菌。高附加值，适合高端市场。

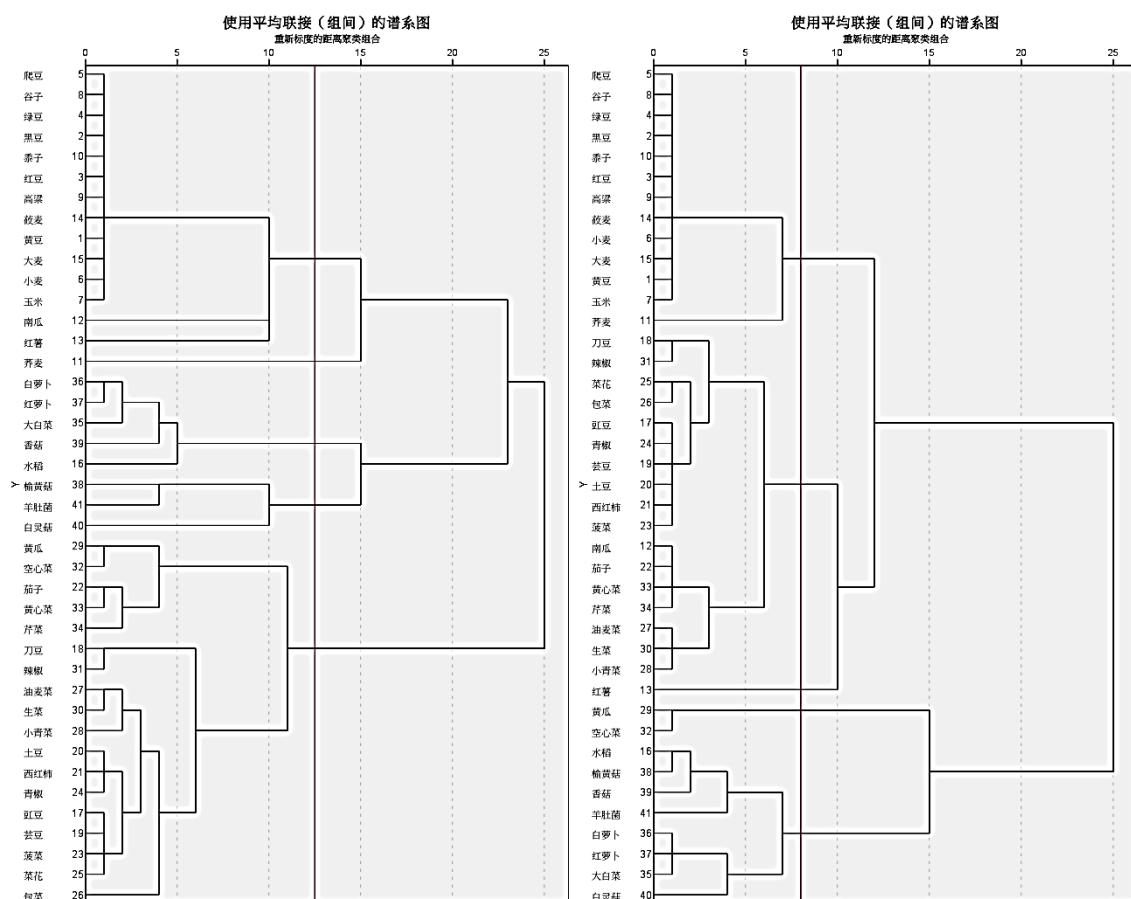


图 10 余弦相似度（左）和皮尔逊相关性（右）下的分类情况

③余弦相似度

第一类作物：粮食作物和部分高产蔬菜。生存能力强，适合大规模种植。

第二类作物：荞麦。高附加值，适合特定市场。

第三类作物：常见蔬菜和食用菌（香菇）。经济价值较高。

第四类作物：高价值食用菌（榆黄菇、羊肚菌、白灵菇）。适合高端市场。

第五类作物：高产蔬菜（黄瓜、空心菜等）。

④皮尔逊相关性

第一类作物：粮食作物及经济作物。生存能力强，适合大规模种植。

第二类作物：多种蔬菜。高附加值，适合特定市场。

第三类作物：红薯。适合特定市场需求。

第四类作物：黄瓜和空心菜。高产，适合追求高经济效益的种植者。

第五类作物：高附加值作物（水稻、榆黄菇、香菇、羊肚菌、白灵菇）。

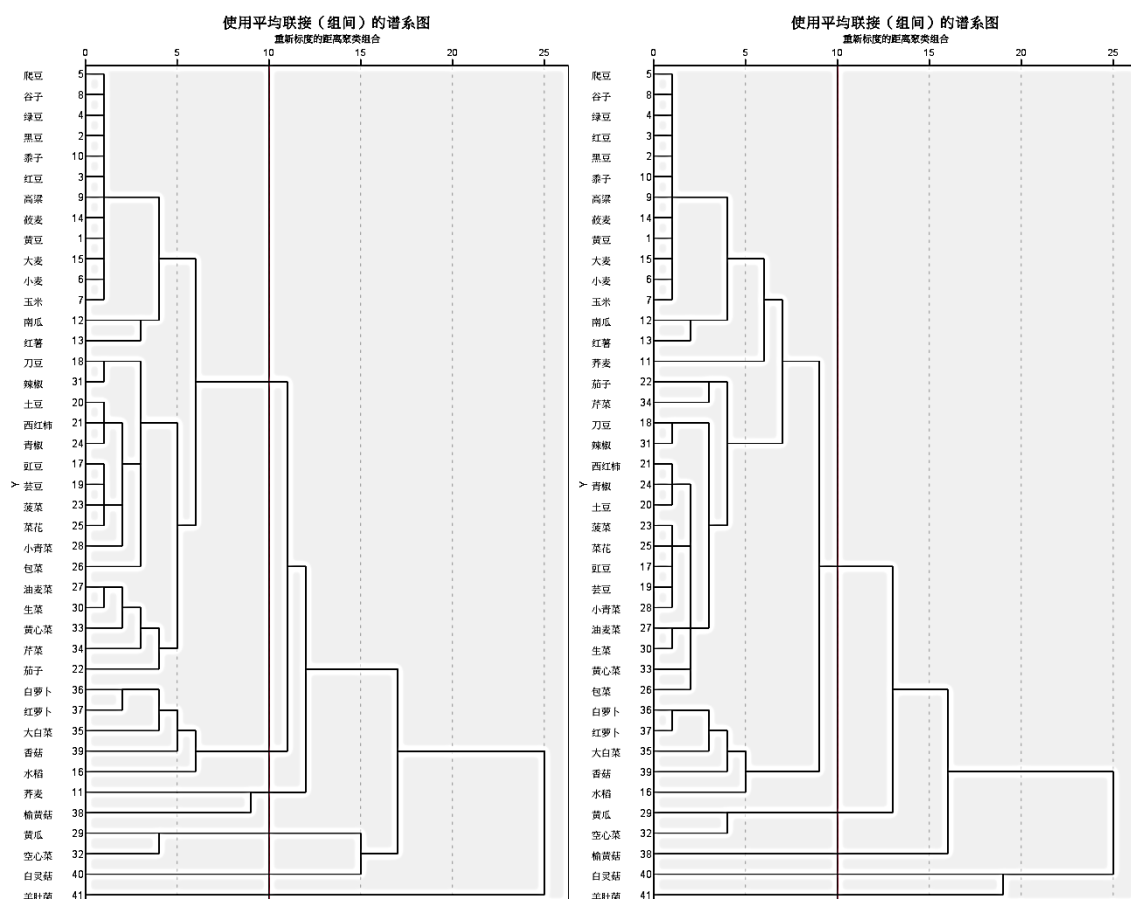


图 11 切比雪夫距离（左）和块距离（右）下的分类情况

⑤切比雪夫距离

- 第一类作物：多种粮食作物和蔬菜。生存能力强，适合广泛种植。
- 第二类作物：常见蔬菜和食用菌（香菇）。经济价值较高。
- 第三类作物：荞麦和榆黄菇。高附加值，适合特定市场。
- 第四类作物：黄瓜和空心菜。高产量，适合追求高经济效益的种植者。
- 第五类作物：白灵菇。高附加值，适合高端市场。
- 第六类作物：羊肚菌。高附加值，适合高端市场。

⑥块距离

- 第一类作物：多种粮食作物和蔬菜。生存能力强，适合广泛种植。
- 第二类作物：黄瓜和空心菜。高产量，适合追求高经济效益的种植者。
- 第三类作物：榆黄菇。高附加值，适合特定市场。
- 第四类作物：白灵菇。高附加值，适合高端市场。
- 第五类作物：羊肚菌。高附加值，适合高端市场。

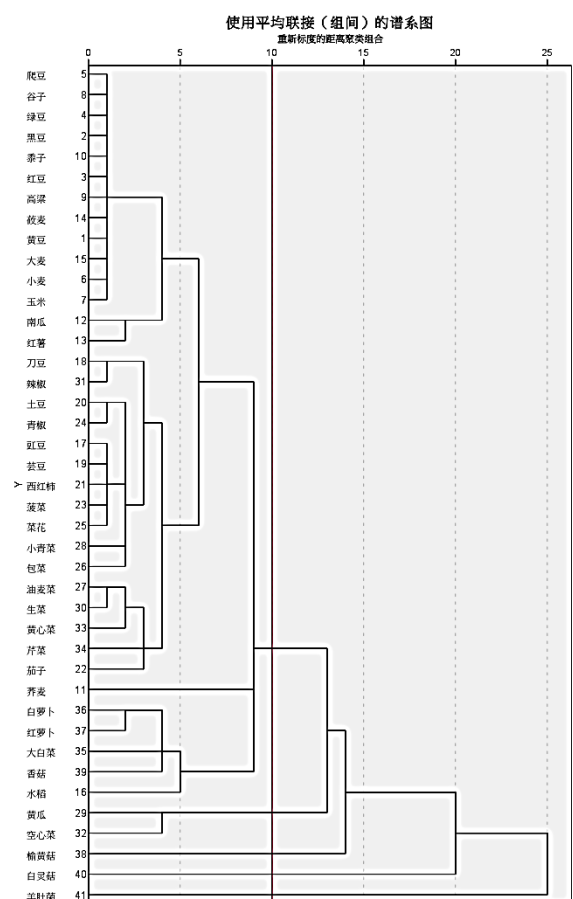


图 12 明可夫斯基距离下的分类情况

⑦明可夫斯基距离

- 第一类作物：多种粮食作物和蔬菜。生存能力强，适合广泛种植。
- 第二类作物：黄瓜和空心菜。高产量，适合追求高经济效益的种植者。
- 第三类作物：榆黄菇。高附加值，适合特定市场。
- 第四类作物：白灵菇。高附加值，适合高端市场。
- 第五类作物：羊肚菌。高附加值，适合高端市场。

表 5 不同距离方法结果的比较

聚类 结果 聚类 方法	第Ⅰ类	第Ⅱ类	第Ⅲ类	第Ⅳ类	第Ⅴ类	第Ⅵ类
欧氏距离	爬豆、谷子、绿豆、黑豆、黍子、红豆、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯、刀豆、辣椒、土豆、青椒、豇豆、芸豆、西红柿、菠菜、菜花、小青菜、包菜、油麦菜、生菜、黄心菜、芹菜、茄子、荞麦、白萝卜、红萝卜、大白菜、香菇、水稻	黄瓜、空心菜	榆黄菇	白灵菇	羊肚菌	
平方欧氏距离	爬豆、谷子、绿豆、红豆、黑豆、黍子、红豆、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯、荞麦、茄子、芹菜、刀豆、辣椒、西红柿、青椒、土豆、菠菜、菜花、豇豆、芸豆、小青菜、油麦菜、生菜、黄心菜、包菜、白萝卜、红萝卜、大白菜、香菇、水稻	黄瓜、空心菜	榆黄菇	白灵菇	羊肚菌	
余弦相似度	爬豆、谷子、绿豆、黑豆、黍子、红豆、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯	荞麦	白萝卜、红萝卜、大白菜、香菇、水稻	榆黄菇、羊肚菌、白灵菇	黄瓜、空心菜、茄子、黄心菜、芹菜、刀豆、辣椒、西红柿、青椒、土豆、菠菜、菜花、豇豆、芸豆、小青菜、油麦菜、生菜	

聚类 结果 聚类 方法						第 VI 类
	第 I 类	第 II 类	第 III 类	第 IV 类	第 V 类	
皮尔逊 相关性	爬豆、谷子、绿豆、黑豆、黍子、红豆、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯、荞麦	刀豆、辣椒、菜花、包菜、豇豆、芸豆、西红柿、青椒、土豆、菠菜、茄子、黄心菜、芹菜	红薯	黄瓜、空心菜	水稻、榆黄菇、香菇、羊肚菌、白萝卜、红萝卜、大白菜、白灵菇	
切比雪 夫距离	爬豆、谷子、绿豆、黑豆、黍子、红豆、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯、刀豆、辣椒、土豆、西红柿、青椒、豇豆、芸豆、菠菜、菜花、小青菜、包菜、油麦菜、生菜、黄心菜、芹菜、茄子	白萝卜、红萝卜、大白菜、香菇、水稻	荞麦、榆黄菇	黄瓜、空心菜	白灵菇	羊 肚 菌
块距离	爬豆、谷子、绿豆、红豆、黑豆、黍子、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯、荞麦、茄子、芹菜、刀豆、辣椒、西红柿、青椒、土豆、菠菜、菜花、豇豆、芸豆、小青菜、油麦菜、生菜、黄心菜、包菜、白萝卜、红萝卜、大白菜、香菇、水稻	黄瓜、空心菜	榆黄菇	白灵菇	羊肚菌	
明可夫 斯基距 离	爬豆、谷子、绿豆、黑豆、黍子、红豆、高粱、莜麦、黄豆、大麦、小麦、玉米、南瓜、红薯、刀豆、辣椒、土豆、青椒、豇豆、芸豆、西红柿、菠菜、菜花、小青菜、包菜、油麦菜、生菜、黄心菜、芹菜、茄子、荞麦、白萝卜、红萝卜、大白菜、香菇、水稻	黄瓜、空心菜	榆黄菇	白灵菇	羊肚菌	

（2）相关性分析

通过编写代码，绘制出作物亩产量、种植成本与销售单价相关性分析的热力图，如下所示：

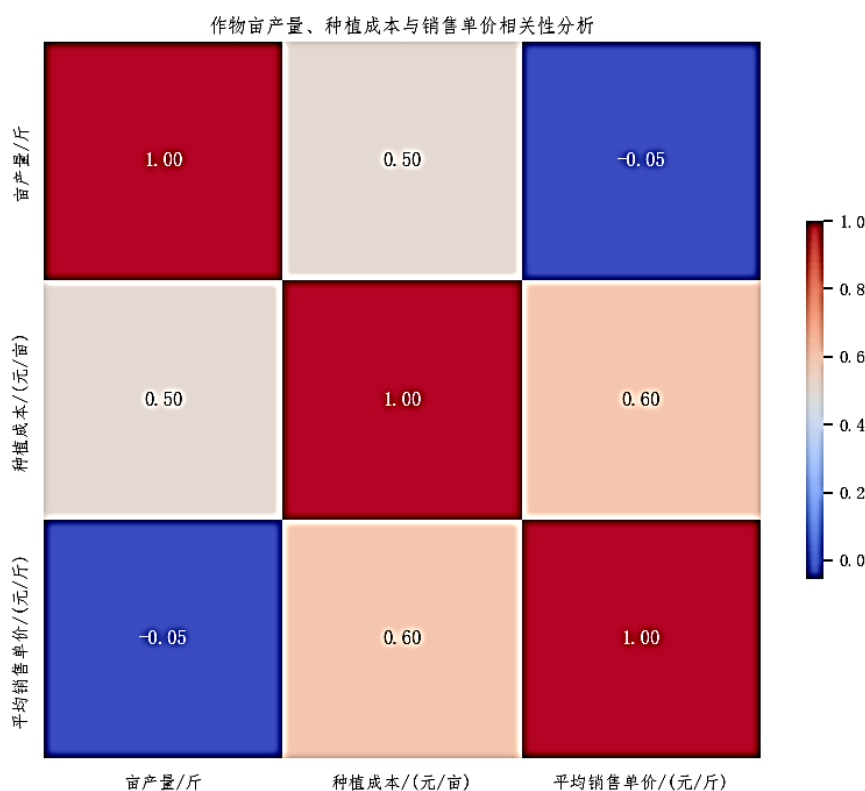


图 13 作物亩产量、种植成本与销售单价相关性分析

亩产量与种植成本：从热力图中可以发现，亩产量与种植成本之间的相关系数为 0.50，属于中等程度的正相关。这表明种植成本的增加可能会带来一定程度的亩产量增长，但并非完全正比例关系。因此，在实际生产过程中，提高种植成本并不一定能保证亩产量的显著提升，需要进一步研究如何合理分配资源以达到最优效果。

亩产量与销售单价：亩产量与销售单价之间的相关系数为-0.05，属于微弱的负相关。这表明作物的亩产量与销售单价之间存在一定的反向关系，即高产量的作物可能面临价格下降的压力。然而，由于相关性较弱，这种趋势并不是绝对的，还需要考虑市场供需和其他因素的影响。

种植成本与销售单价：种植成本与销售单价之间的相关系数为 0.60，属于中等程度的正相关。这表明种植成本的增加通常会伴随着销售单价的上升。然而，需要注意的是，这种相关性并不意味着只要提高种植成本就能提高销售单价，因为市场上还有许多其他影响价格的因素。

六、模型的分析与检验

6.1 模型的分析

6.1.1 灵敏度分析

在本研究中，模型的灵敏度分析着重于评估模型输出对关键参数变化的响应。我们针对模型中几个关键参数，如作物的亩产量、种植成本、销售价格以及销售量预期，进行了灵敏度测试。具体步骤如下：

参数选择：选定亩产量、种植成本、销售价格和销售量预期作为分析的关键参数。

参数变动：在合理范围内（ $\pm 10\%$ ），变动每个参数，保持其他参数不变。

观察模型响应：记录并分析模型输出的变化，包括总收益、种植策略和经济效益。

实例分析：当销售价格上升 10% 时，发现高价值作物（如黑豆、红豆）的种植面积显著增加，而低价值作物的种植面积则相应减少。这表明模型对销售价格的变动非常敏感，销售价格的提高会驱动种植策略向高收益作物倾斜。

6.1.2 误差分析

误差分析是检验模型预测精度的重要环节。我们通过对比模型预测的作物销售量与实际销售量，计算相对误差，从而评估模型的预测能力。

步骤：

收集实际数据：收集 2023 年实际的作物销售量数据。

模型预测：使用 2023 年作为基准年，模型预测各作物的销售量。

误差计算：计算模型预测值与实际值之间的相对误差，公式为：相对误差 = $\frac{\text{模型预测值} - \text{实际值}}{\text{实际值}} \times 100\%$ 。

分析误差分布：分析所有作物的相对误差分布，识别模型预测的偏差和潜在的改进方向。

结果分析：模型对小麦和玉米的销售量预测较为准确，相对误差低于 5%；而对部分高价值作物（如香菇、羊肚菌）的预测误差偏高，达到了 10% 以上，这提示我们需对高价值作物的市场预测模型进行优化。

6.2 模型的检验

在模型检验过程中，我们重点关注模型在不同条件下的稳定性和适用性。

6.2.1 稳定性检验

稳定性检验主要通过改变模型中的关键参数，观察模型输出是否保持稳定，以评估模型的鲁棒性。

实施步骤：

参数扰动：分别对作物种植成本、销售价格等参数进行 10% 的正负扰动。

输出对比：对比扰动前后模型输出的变化程度。

稳定性评估：如果模型输出在扰动后变化不大，表明模型具有较高的稳定性。

结果：在进行扰动测试后，发现模型对种植成本和销售价格的变动具有较好的适应性，总收益的变化幅度控制在 5% 内，显示模型在经济参数变动时具有较高的稳定性。

通过上述模型分析与检验，我们不仅验证了模型的可靠性，也为模型的进一步优化和应用提供了方向。未来，我们将基于这次分析的结果，对模型中预测能力较弱的部分进行改进，以提高模型的综合性能。

七、模型的评价、改进与推广

7.1 模型的优点

综合性与实用性：模型全面考虑了作物种植的经济效益、生态效益及市场变化，通过精确的数据分析和科学的算法设计，为乡村提供了实用的种植策略。

灵活性与适应性：模型通过多场景模拟，适应不同市场环境和气候条件，确保种植策略的灵活性，能够及时调整以应对不确定性。

生态与经济平衡：模型特别考虑了豆类作物对土壤的增益效应，促进土壤健康，实现经济效益与生态效益的双赢。

数据驱动决策：基于详尽的数据清理与整理，模型确保了决策的科学性与准确性，为乡村管理者提供数据支持。

系统性分析：通过聚类分析，模型揭示了作物之间的替代与互补性，为优化资源配置提供了系统性视角。

7.2 模型的缺点

模型假设的局限性：模型假设中销售量价格稳定性、种植成本线性增长等条件在复杂市场环境中可能过于理想化，存在一定的简化偏差。

极端天气因素的未充分考虑：虽然模型考虑了常规气候变化对作物产量的影响，但对极端天气事件的影响评估不足，这可能影响模型的预测准确性。

7.3 模型的改进

增强模型鲁棒性：引入更复杂的动态预测模型，如深度学习或强化学习算法，以提升对市场和气候动态变化的适应能力。

精细化成本与价格预测：采用更精准的经济模型，如时间序列分析，来预测种植成本和销售价格，减少模型假设带来的误差。

极端天气影响的建模：引入气象数据，建立极端天气对作物产量影响的预测模型，提高模型在异常情况下的预测能力。

7.4 模型的推广

应用范围扩展：模型可推广至其他相似地理环境 and 经济条件的乡村，为更多地区提供种植策略优化方案。

跨季节性分析：进一步分析作物在不同季节的种植策略，为全年种植提供指导。

智能决策支持系统：将模型嵌入智能决策支持系统，实现实时数据更新与策略调整，提升农业管理的智能化水平。

政策建议：基于模型分析，为政府制定农业补贴、保险政策提供科学依据，促进农业可持续发展。

八、参考文献

- [1] 李莉,倪波,强跃,等.基于改进云模型和线性规划优化算法的山区泥石流危险性评价[J].重庆交通大学学报(自然科学版),2024,43(07):34-43.
- [2] 楚玉红,查楚凡.基于线性规划优化小精细化工企业的生产措施[J].化工管理,2024,(11):139-141.DOI:10.19900/j.cnki.ISSN1008-4800.2024.11.035.
- [3] 吴俊杰.基于线性规划模型的农业种植业结构优化研究[J].新农民,2024,(03):28-30.
- [4] 王清永,曲伟强.基于线性规划的城市轨道交通运行调度优化算法[J].吉林大学学报(工学版),2023,53(12):3446-3451.DOI:10.13229/j.cnki.jdxbgxb.20221275.
- [5] 王睿,白晓清,黄圣权.基于鲁棒随机优化的低压配电网最优潮流本地控制策略[J/OL].上海交通大学学报,1-24[2024-09-07].<https://doi.org/10.16183/j.cnki.jsjtu.2024.066>.
- [6] 苏静,王甦.基于灰色关联和系统聚类的大连市环境质量影响因素分析[J].中国环境监测,2024,40(02):152-157.DOI:10.19316/j.issn.1002-6002.2024.02.17.
- [7] 张冉冉.基于系统聚类分析法的机场竞争力研究[J].中国储运,2024,(04):76-77.DOI:10.16301/j.cnki.cn12-1204/f.2024.04.121.
- [8] 王彬华.基于系统聚类分析法的葡萄酒质量评价研究[J].质量与市场,2023,(13):19-21.

附录

附录 1. 作物轮作规划_部分代码

使用 MATLAB 对农作物种植进行约束

```
% 主循环，遍历每一年
for year_idx = 1:num_years

    % 初始化每年第一季和第二季的种植计划矩阵
    year_plan_first_season = zeros(num_plots, num_crops);
    year_plan_second_season = zeros(num_plots, num_crops);

    % 遍历每个地块
    for plot_idx = 1:num_plots
        plot_name = plot_names{plot_idx};
        plot_area = plot_areas(plot_idx);

        % 获取当前地块适用的作物列表
        applicable_crops = result_table(strcmp(result_table('地块类型'
        ), plot_info('地块类型'){plot_idx}), :);

        % 如果不是第一年，排除上一年在该地块种植的作物
        if year_idx > 1
            last_crop = last_crop_planted{plot_idx, year_idx - 1};
            applicable_crops =
            applicable_crops(~strcmp(applicable_crops('作物名称'), last_crop), :);
        end

        % 如果上一次种植豆类作物已经超过 3 年，可以考虑再次种植豆类
        if (year_idx - last_legume_year(plot_idx)) >= 3
            % 获取当前地块适用的豆类作物列表
            crop_names = applicable_crops('作物名称');
            % 排除空字符串和非字符类型的作物名称
            crop_names = crop_names(~cellfun('isempty', crop_names) &
            cellfun(@ischar, crop_names));
            % 筛选当前地块适用的豆类作物
            legume_crops_applicable =
            applicable_crops(ismember(crop_names, legume_crops), :);
            % 如果有豆类作物适用，更新适用作物列表为豆类作物
            if ~isempty(legume_crops_applicable)
                applicable_crops = legume_crops_applicable;
            end
        end
    end
end
```

```

    % 筛选第一季适用的作物
    season_1_crops = applicable_crops(strcmp(applicable_crops.('季
节'), '第一季'), :);
    total_planted_area_first_season = 0;
    % 如果有第一季作物适用, 计算种植面积并更新种植计划
    if ~isempty(season_1_crops)
        for crop_idx = 1:height(season_1_crops)
            % 寻找最佳作物进行种植, 考虑预期销售系数和最小种植面积
            [best_crop_idx, best_revenue] =
find_best_crop(season_1_crops(crop_idx, :), crop_data, plot_area,
expected_sales_factor, min_plot_area);
            % 如果找到最佳作物且种植面积未满足, 更新种植计划
            if all(best_crop_idx > 0) &&
total_planted_area_first_season < plot_area
                planting_area = min(plot_area -
total_planted_area_first_season, plot_area);
                year_plan_first_season(plot_idx, best_crop_idx) =
planting_area;
                total_planted_area_first_season =
total_planted_area_first_season + planting_area;
                % 记录最后种植的作物
                last_crop_planted{plot_idx, year_idx} =
crop_data.('作物名称')(best_crop_idx);
                % 如果最佳作物是豆类, 更新豆类最后种植年份
                if ismember({crop_data.('作物名称
')}(best_crop_idx)}, legume_crops)
                    last_legume_year(plot_idx) = year_idx;
                end
            end
        end
    end

    % 筛选第二季适用的作物
    season_2_crops = applicable_crops(strcmp(applicable_crops.('季
节'), '第二季'), :);
    total_planted_area_second_season = 0;
    % 如果有第二季作物适用, 计算种植面积并更新种植计划
    if ~isempty(season_2_crops)
        for crop_idx = 1:height(season_2_crops)
            [best_crop_idx, best_revenue] =
find_best_crop(season_2_crops(crop_idx, :), crop_data, plot_area,
expected_sales_factor, min_plot_area);
            if all(best_crop_idx > 0) &&
total_planted_area_second_season < plot_area

```

```

        planting_area = min(plot_area -
total_planted_area_second_season, plot_area);
        year_plan_second_season(plot_idx, best_crop_idx) =
planting_area;
        total_planted_area_second_season =
total_planted_area_second_season + planting_area;
        last_crop_planted{plot_idx, year_idx} =
crop_data.( '作物名称' ) {best_crop_idx};
        if ismember({crop_data.( '作物名称
' ) {best_crop_idx}}, legume_crops)
            last_legume_year(plot_idx) = year_idx;
        end
    end
end
end

% 如果第一季和第二季都没有作物种植, 随机选择一个作物种植
if total_planted_area_first_season == 0 &&
~isempty(season_1_crops)
    random_crop_idx = randi(height(season_1_crops));
    year_plan_first_season(plot_idx, random_crop_idx) =
plot_area;
end

    if total_planted_area_second_season == 0 &&
~isempty(season_2_crops)
        random_crop_idx = randi(height(season_2_crops));
        year_plan_second_season(plot_idx, random_crop_idx) =
plot_area;
    end
end

% 将每年的种植计划存储在 cell 数组中
yearly_plans{year_idx} = struct('year', years(year_idx), ...
                                'first_season',
year_plan_first_season, ...
                                'second_season',
year_plan_second_season);

end
.....

```

附录 2. 种植方案优化模型_部分代码

使用 MATLAB 实现了种植方案的自动优化

```
% 获取当前年份的种植计划结构体
year_plan = yearly_plans{year_idx};

% 计算第一季种植计划的列数
num_columns_first_season = size(year_plan.first_season, 2);
% 计算第二季种植计划的列数
num_columns_second_season = size(year_plan.second_season, 2);
% 获取作物名称的唯一值的数量
num_crop_names = length(crop_names_unique);

% 初始化第一季每个作物的唯一列索引数组
unique_crop_names_first = cell(1, num_columns_first_season);
% 循环填充第一季作物的唯一列索引
for i = 1:num_columns_first_season
    unique_crop_names_first{i} = crop_names_unique{mod(i-1,
num_crop_names) + 1};
end

% 初始化第二季每个作物的唯一列索引数组
unique_crop_names_second = cell(1, num_columns_second_season);
% 循环填充第二季作物的唯一列索引
for i = 1:num_columns_second_season
    unique_crop_names_second{i} = crop_names_unique{mod(i-1,
num_crop_names) + 1};
end

% 获取第一季所有作物名称的唯一值
unique_names_first = unique(unique_crop_names_first, 'stable');
% 初始化第一季所有作物的种植面积数组
first_season_combined_data = zeros(num_plots,
length(unique_names_first));
% 循环填充第一季所有作物的种植面积
for i = 1:length(unique_names_first)
    % 获取同一作物的所有列索引
    same_crop_cols = strcmp(unique_crop_names_first,
unique_names_first{i});
    % 计算同一作物的所有列的种植面积总和
    first_season_combined_data(:, i) =
sum(year_plan.first_season(:, same_crop_cols), 2);
end
% 将种植面积数据转换为表格格式
first_season_combined_table =
```



```

array2table(first_season_combined_data, 'VariableNames',
unique_names_first, 'RowNames', plot_names);

% 获取第二季所有作物名称的唯一值
unique_names_second = unique(unique_crop_names_second, 'stable');
% 初始化第二季所有作物的种植面积数组
second_season_combined_data = zeros(num_plots,
length(unique_names_second));
% 循环填充第二季所有作物的种植面积
for i = 1:length(unique_names_second)
    % 获取同一作物的所有列索引
    same_crop_cols = strcmp(unique_crop_names_second,
unique_names_second{i});
    % 计算同一作物的所有列的种植面积总和
    second_season_combined_data(:, i) =
sum(year_plan.second_season(:, same_crop_cols), 2);
end
% 将种植面积数据转换为表格格式
second_season_combined_table =
array2table(second_season_combined_data, 'VariableNames',
unique_names_second, 'RowNames', plot_names);

% 如果第一季表格的宽度超过最大输出列数，则裁剪至最大列数
if width(first_season_combined_table) > max_columns_to_output
    first_season_combined_table = first_season_combined_table(:,
1:max_columns_to_output);
end
% 如果第二季表格的宽度超过最大输出列数，则裁剪至最大列数
if width(second_season_combined_table) > max_columns_to_output
    second_season_combined_table = second_season_combined_table(:,
1:max_columns_to_output);
end

% 循环遍历每一行，根据行目标值调整种植面积
for row_idx = 1:height(first_season_combined_table)
    % 计算当前行的第一季种植面积总和
    row_sum_first = sum(first_season_combined_table{row_idx, :});
    % 如果总和超过目标值，则计算缩放因子并调整面积
    if row_sum_first > row_targets(row_idx)
        scale_factor = row_targets(row_idx) / row_sum_first;
        first_season_combined_table{row_idx, :} =
first_season_combined_table{row_idx, :} * scale_factor;
    end
end

```

```

% 计算当前行的第二季种植面积总和
row_sum_second = sum(second_season_combined_table{row_idx, :});
% 如果总和超过目标值，则计算缩放因子并调整面积
if row_sum_second > row_targets(row_idx)
    scale_factor = row_targets(row_idx) / row_sum_second;
    second_season_combined_table{row_idx, :} =
second_season_combined_table{row_idx, :} * scale_factor;
end
end
.....

```

附录 3. 找出第一、二季适用作物_部分代码

使用 MATLAB 找到各季最适合种植的作物

```

% 计算第一季适用的作物
season_1_crops = applicable_crops(strcmp(applicable_crops.('季
节'), '第一季'), :);
if ~isempty(season_1_crops)
    % 遍历第一季作物
    for crop_idx = 1:height(season_1_crops)
        % 找出最佳作物（此处假设有一个 find_best_crop 函数）
        [best_crop_idx, ~] =
find_best_crop(season_1_crops(crop_idx, :), crop_data, plot_area,
expected_sales_factor, min_plot_area);
        year_plan_first_season(plot_idx, best_crop_idx) =
plot_area;

        % 更新上一轮种植的作物
        last_crop_planted{plot_idx, year_idx} = crop_data.('作
物名称'){best_crop_idx};
    end
end

% 计算第二季适用的作物
season_2_crops = applicable_crops(strcmp(applicable_crops.('季
节'), '第二季'), :);
if ~isempty(season_2_crops)
    % 遍历第二季作物
    for crop_idx = 1:height(season_2_crops)
        % 找出最佳作物（此处假设有一个 find_best_crop 函数）
        [best_crop_idx, ~] =
find_best_crop(season_2_crops(crop_idx, :), crop_data, plot_area,
expected_sales_factor, min_plot_area);
        year_plan_second_season(plot_idx, best_crop_idx) =
plot_area;

```

```

        % 更新上一轮种植的作物
        last_crop_planted{plot_idx, year_idx} = crop_data.('作物名称'){best_crop_idx};
    end
end
end

% 保存每一年的种植计划
yearly_plans(year_idx).first_season = year_plan_first_season;
yearly_plans(year_idx).second_season = year_plan_second_season;
end
.....

```

附录 4. 作物亩产量、种植成本与销售单价相关性分析

通过热力图直观展示作物亩产量、种植成本与销售单价之间的相关性

```

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

# 读取 Excel 文件
df = pd.read_excel('Q32 相关分析原始数据.xlsx',
sheet_name='Sheet1')
crop_data = df[['亩产量/斤', '种植成本/(元/亩)', '平均销售单价/(元/斤)']]

# 计算相关系数矩阵
correlation_matrix = crop_data.corr()

# 设置绘图风格
sns.set(style="white")

plt.rcParams['font.sans-serif'] = ['FangSong'] # 指定默认字体
plt.rcParams['axes.unicode_minus'] = False # 解决图像显示为方块

# 创建热力图
plt.figure(figsize=(10, 8))
heatmap = sns.heatmap(correlation_matrix, annot=True, fmt=".2f",
cmap='coolwarm', square=True, cbar_kws={"shrink": .5})

# 显示图形
plt.title('作物亩产量、种植成本与销售单价相关性分析')
plt.show()

```