基于遗传算法求解单目标规划制定蔬菜补货定价策略

摘　要

在为生鲜商超制定蔬菜补货与定价策略目标时，为使商超尽可能降低损失、提高利润，本文建立了斯皮尔曼分析模型、混沌时间序列预测模型、非线性规划模型，并用遗传算法（GA）求解制定了补货定价策略。

首先进行异常值剔除、之后用三次样条插值对缺失数据进行补全，为后续模型建立提供数据基础。

针对问题一，研究了单品之间、品类之间销售量分布规律以及相互关系。本文使用 MATLAB 软件进行了数据**可视化处理**，将 6 大品类、30 种单品的销量与时间的关系绘制成了曲线图，并从波动程度、周期性、日销总量等分布规律予以描述。接着本文使用 SPSS 绘制矩阵散点图看出两变量之间的非线性，**Q-Q 图检验**发现其不符合正态分布，据此决定采用 **Spearman 相关性分析**求出品类之间、单品之间销量的**相关系数矩阵**量化分析其相互关系。

针对问题二，研究了销量与定价的关系并给出未来七天品类补货量与定价策略。本文采用**多项式函数拟合法**确定了蔬菜品类总销量与成本加成定价的函数对应关系，拟合优度达 **0.963**，将单品损失率、进货成本采用**加权平均**法计算出品类的损失率与进货成本，考虑到成本随时间变化呈现明显的非线性关系，本文用**混沌时间序列**预测出 1-7 日各品类蔬菜的进货成本。接着利用进货量、销售定价作为决策变量，商超获取利润为目标函数建立了一般化的**单目标规划模型**，利用一定的变换技巧，将模型化为标准的二次型，使用 MATLAB 软件求解出在未来七日每种蔬菜品类的定价与进货量，此策略下七日累计总收益可达 **4362.19 元。**

针对问题三，研究了 7 月 1 日最佳单品补货量与定价策略。本文统计出 6 月 24-30 日的可售品种共计 49 个，沿用第二问的拟合函数作为单品销量与定价的关系，采用**混沌时间序列**预测出单品的成本，引入**打折优惠**行为以促进销量。利用单品进货量、成本加成定价作为决策变量，单品控制 27~33 个、最小陈列量 2.5 千克以及实际销量低于补货量为主作为约束条件，当日利润最大为目标函数，建立**单目标规划模型**，利用非线性规划理论与**遗传算法（GA）**进行求解。值得注意的是，本题充分利用第一问求解的相关性系数大小，指导遗传算法**交叉优先级，**最终 49 个单品经过遗传算法淘汰后保留了 **31 个单品**，并给出了这些单品对应的建议定价、建议补货量、打折优惠力度等，并计算出执行此项策略后预计商超在 7 月 1 日的收益为 **623.02 元**。

针对问题四，研究了其他因素对补货定价策略产生的影响。本文基于生活经验，决定进一步统计如下数据：

（1）统计商超进货成本价与进货量之间的关系，考虑进货量大优惠

（2）各蔬菜单品的保质期天数，基于保质期修正补货量

（3）商超的冷藏保鲜成本与库存容量，基于保鲜成本与库存修正补货量

（4）统计不同天气影响人们的采购量与偏好的变化，基于天气预报修正补货量

最后我们对模型的鲁棒性进行了检验，发现模型具有较好的鲁棒性。同时我们也对模型的优缺点进行了评价。

关键词：补货定价策略，混沌时间序列，单目标规划，遗传算法，函数拟合

一、 问题重述

1.1 问题背景

随着人民生活水平不断提高，人们对于食品的安全性、新鲜度有着越来越高的要求。以蔬菜类为例，与一般商品不同，该类易逝生鲜产品的保质期大都较短，且产品品质往往也会随销售时间的推移而逐渐递减，而且该类产品往往会存在较为明显的时令效应，这直接导致生鲜商超需要严格地按日需供应产品，既需要考虑当日蔬菜的销售空间、需求量、补货量与售价，又得兼顾时令等因素带来的季节性供需变化，如此一来为商品进货并定价就变得极为繁琐，稍有不慎便会造成经济损失和食物浪费。由此可见，研究蔬菜类商品的自动定价与补货策略的问题十分具有现实意义与经济效益。为此本文需解决如下问题：

1.2 问题重述

问题一：要求探究不同蔬菜品类的销售量之间以及不同蔬菜单品之间的相关性，并分析出各品类以及各单品销售量基于时间上的分布规律与基于销量上相互关系。

问题二：仅以商品类别为单位制定补货计划，要求分析出六大类蔬菜品类各自的销售总量与成本加成定价的关系，同时利用该关系为商超制定出 2023 年 7 月 1-7 日七天利润最大时的日补货量与定价策略。

问题三：实际售卖过程中，综合考虑到商超售卖空间的有限性、陈列菜品的最小量、可售单品总数、可售品种的有限性等符合现实问题的约束条件，要求给出 7 月 1 日当天的单品补货量和定价策略，尽可能满足市场需求并且盈利最大化。

问题四：为了更好地指定补货与定价策略，请为商超提出其他数据的采集建议，并给出这些数据对于优化策略的意义。

二、 问题分析

2.1 问题一的分析

在问题一中，确定相关关系的主体有两个，其一是不同品类之间销售量的关系，其二是不同单品之间销售量的关系。显然，销售量又与时间相关，即需要确保每一类菜品在进行销量相关性比较时需要在同一时间维度上，否则没有实际意义。问题一的问题也有两个，即分布规律描述和相关性分析，对于前者，可以用 MATLAB 将数据可视化，结合生活经验与数据特征对其进行描述，对于后者则可以采用相关的模型量化分析其相关性强弱、正负等。考虑到蔬菜销售量往往与菜品成熟期有关，同时为简化数据处理的难度，本文将 3 年的数据指标按照每年 4 个季度共划分成 12 个季度，将这 12 个季度作为每一类蔬菜品类的以及每一种单品的 12 个特征指标。将每个季度中日销量求和作为每个季度的总销量。采用斯皮尔曼（Spearman）相关性分析，对 6 类菜品和 30 个单品分别进行求解，最终得到 6 类菜品之间及 30 个单品之间的相关系数矩阵，由此便可确定单品之间、品类之间的相关性。

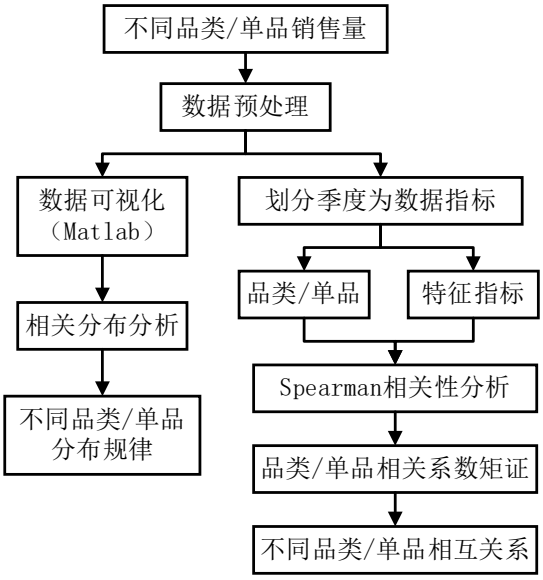


图 2.1 问题 1 解决流程图

2.2 问题二的分析

在问题二中，题目首先要求分析出各蔬菜品类与销售定价的关系，由于题目是以品类为单位，因此每个单品便可不予区分，可将 T 日该品类下所有单品销量S_{SI}^{T}求和作为品类销量S_{CN}^{T}，再将该类菜品下属 T 日所有单品定价P_{SI}^{T}取加权平均的方式作为品类定价P_{CN}^{T}。品类定价作为横坐标，品类销量作为纵坐标，即可做出P_{SI}^{T}-S_{SI}^{T}的散点图，最后采用函数拟合的方法便可得到销量与定价的函数关系。随后题目要求求出未来一周使得商超收益最大的补货量与定价策略，可列出收益表示式如下：

(1)收益=定价*销量-进货成本*进货量
(2)进货量=销量＋损失量
(3)销量=f(定价)

未来七天的各品类进货成本本文可采用混沌时间序列预测模型进行预测，损失则按照过往经验利用加权平均算得。进货量、销售定价均取有记录以来合理的最大值和最小值作为上下限，对函数值进行必要限制。显然，收益的计算式可最终化简成为一个补货量为变量，收益为因变量的标准二次型函数，使用 Lingo 求解得商超收益最大时的销量和定价。

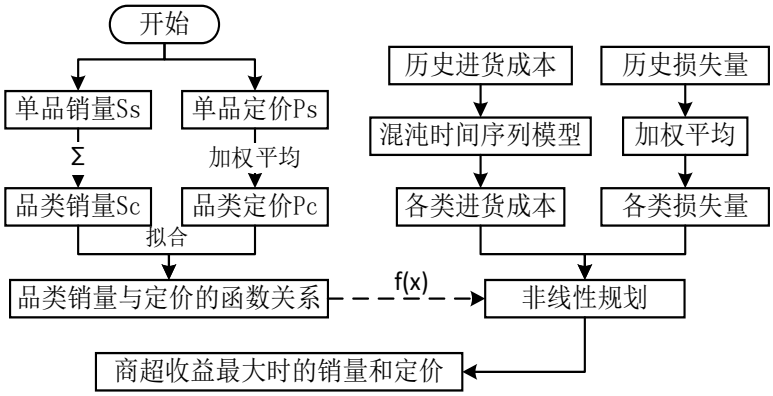


图 2.2 问题二解决流程图

2.3 问题三的分析

问题三中，题目增加了销售空间的限制，更加符合实际情况，这就需要商超在有限的空间内，对与销售菜品做出取舍，销售获利较大的菜品。同时题目还规定了最小陈列量，即补货量的下限。这是一个典型的优化类问题，本文采用将收益函数作为目标函数使其取得最大值。其余条件作为约束条件。用非线性规划数学理论与遗传算法进行求解，收益函数作为适应度函数，将每一个单品作为个体，共计 68 个，再参考第一问与第二问的得到的菜品之间的相关性，具体定义交叉形式与变异概率。最终得到商超收益最大时的补货量与定价策略。

2.4 问题四的分析

在问题四中，题目要求为更好地指定补货与定价策略，商超还需要采集哪些数据使得补货定价策略效果更好。本题的核心目标是优化补货量与定价。而基于第二问中求解的定价与销量关系函数$f(x)$，本文先从补货量入手进行分析。首先，商超在进货时，可能会因进货量不同而获得程度不同的优惠折扣，因此对于有些保质期较长的蔬菜可以一次性大量进货，以获得优惠折扣。其次如需多余进货还需要确保商超有额外的库存容量，甚至有些蔬菜进行简单的冷藏保鲜处置会进一步提升经济效益，而这可能需要花费额外的冷藏成本。知道这些数据后，可以在本文的基础上构建一个库存模型，将库存量引入到非线性规划模型的约束条件中，进一步修正补货量，以使利润达更大。最后，天气因素也是影响人们采购习惯的重要方面。商超还应统计出不同天气情况下人们的采购需求变化，这样可以根据当日天气预报，调整补货策略。

三、 模型假设

1. 假设不存在自然灾害导致蔬菜供应短缺或价格变化
2. 计算盈利时不考虑运费、搬运费等其他费用
3. 假设流量均为自然流量，有持续消费者访问且消费者都是经济人
4. 假设销量仅与价格相关，不考虑其他影响因素
5. 假设不同的包装形式不影响同一种蔬菜的销量

6. 假设补货量只考虑销量与损失量，不考虑大批量进货等情况

四、 符号说明

符号	说明	单位
W	商超获得的利润	元
C_{CN}^{T}	品类N在T天的进货成本	元/千克
C_{SI}^{T}	单品I在T天的进货成本	元/千克
B_{CN}^{T}	品类N在T天的补货量	千克
B_{SI}^{T}	单品I在T天的补货量	千克
S_{CN}^{T}	品类N在T天的销量	千克
S_{SI}^{T}	单品I在T天的销量	千克
L_{CN}	品类N的损失率	无
$f(x)$	销量与售价的关系函数	千克
P_{CN}^{T}	品类N在T天的售价	元/千克
P_{SI}^{T}	单品I在T天的售价	元/千克
F_{SI}	单品I的优惠折扣	无

五、 数据预处理

各品类蔬菜乃至单品销量的预测成功率主要取决于过去该品类蔬菜销售量数据的质量和数量，由于可能受到停业整顿、商品断供、机器故障、特殊情况、大批量订单等不可控因素的影响，销售流水明细所提供的数据并非一定是客观全面的，采集的数据有可能会存在异常、缺失等情况，需要剔除异常的数据，补全缺失的数据。如果用这些数据来预测，会导致预测成功率下降，数据预处理步骤如下：

5.1 异常值剔除

我们绘制了图 1~图 6 即茄类、花菜类、辣椒类、水生根茎类、花叶类六大品类每日销量曲线图，从图 1 可以看出，个别茄类当日销量数据异常（销量远超正常补货量）或许是存在大批量进货现象或者机器故障，但由于我们的目标是预测平常的定价与补货策略，这些异常数据可能会对整体结果产生影响，从而减小模型的精度，甚至导致预测结果完全偏离实际值，因此我们进行异常值剔除。

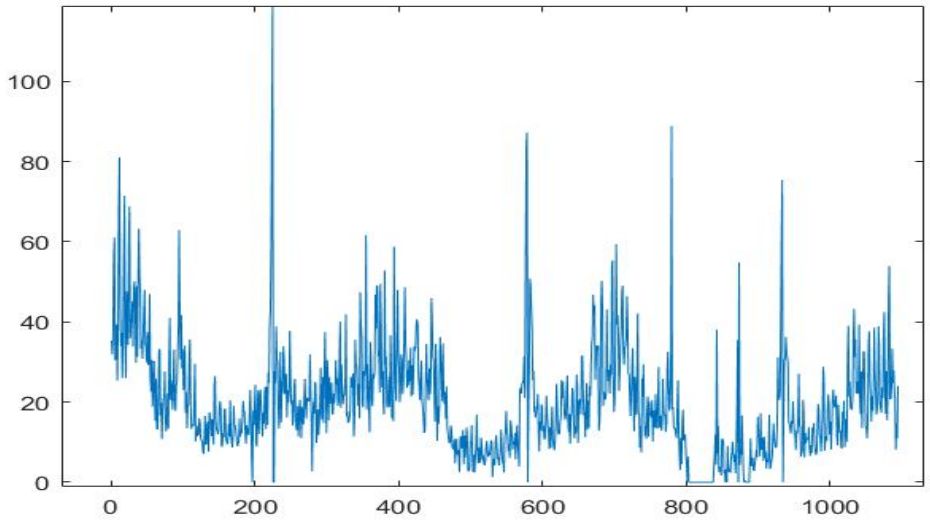


图 5.1 茄类每日销量原始数据曲线图

考虑到样本数据的异常值会影响预测精度，本文异常值剔除决定采用拉依达准则，即 3σ 准则：对于 1000 天的日销量的原始数据，当偏差大于 3σ 的时候，判定该数据为异常值，应被及时剔除，σ 的计算公式如下：

$$\sigma=\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} \tag{1}$$

其中，$x_i(i=1\sim1000)$是指 1000 天中六大类菜品销售总量的原始数据，$\bar{x}$ 是指六类菜品的销量平均值，样本容量(时间跨度)为 100 天。判定数据为异常值的准则为：

$$|x_i-\bar{x}|>3\sigma \tag{2}$$

因此，当数据范围满足(2)式时，将其剔除，剩下的不为 0 的数据即认为是正常值，1095 个数据中共含有 123 个超范围数据和 35 个值为 0 的数据共计 158 个异常值，全部剔除后，总共剩下 937 天的正常日销量数据。

5.2 缺失值补充

在处理完异常数据后，产生了一些列的缺失数据，本文决定采用三次样条插值法对其进行补全操作。取缺失数据前后各取 50 天数据，使用 MATLAB 工具箱中的三次样条插值，其中三次样条函数如下：

$$s_3(x)=a_0+a_1x+\frac{a_2}{2!}x^2+\frac{a_3}{3!}x^3+\sum_{j=1}^{n-1}\frac{\beta_j}{3!}(x-x_j)_+^3\in S_P(\Delta,3) \tag{3}$$

$$(x-x_j)_+^3=\begin{cases}(x-x_j)^3, x\geqslant x_j\\ 0,\ \ x<x_j\end{cases},(j=1,2,...,n-1) \tag{4}$$

由于$s_3(x)\in S_P(\Delta,3)$中含有 n+3 个待定系数，故需要 n+3 个插值条件，已知插值节点x_i和相应函数$f(x_i)=y_i(i=0,1,2,...,n)$，共有 n+1 个条件，剩余两个则采用非扭结条件。

5.3 预处理后的数据

经过异常值剔除、数据补全之后得到共计 1095 个数据（2020 年 7 月 1 日起至 2023 年 60 月 30 日截至），附件图 1~图 6 分别是预处理后的茄类、根茎类、辣椒类三年日销量数据曲线图，这里篇幅限制仅展示两幅图片。

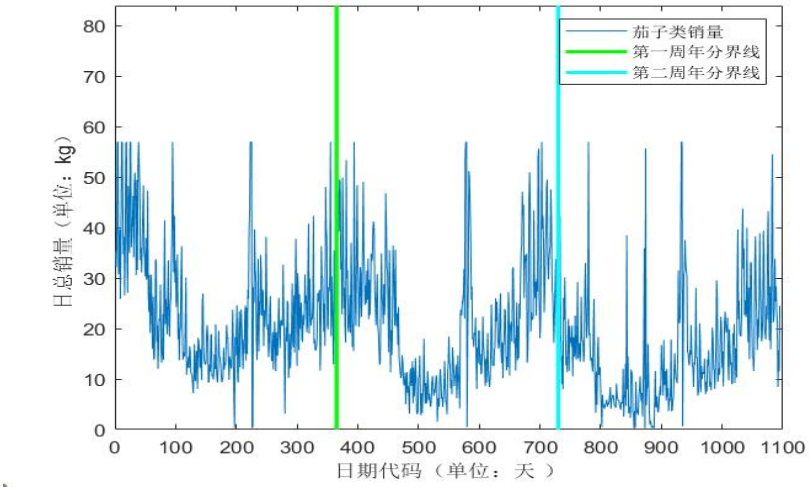


图 5.2 预处理后茄子类每日销量曲线图

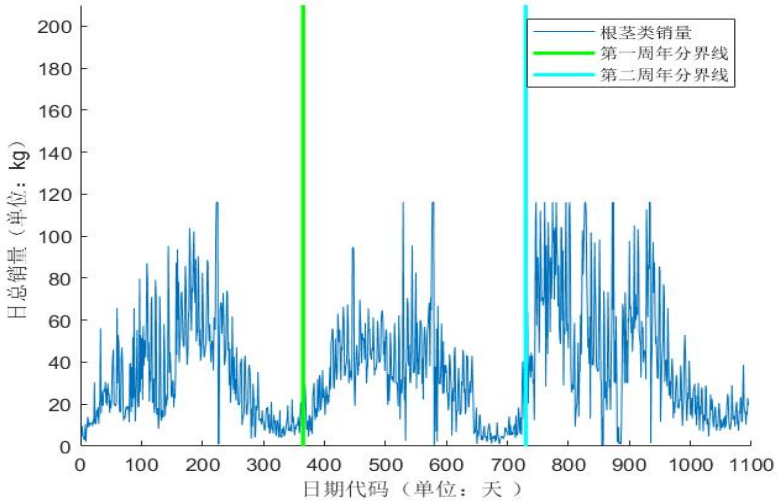


图 5.3 预处理后根茎类每日销量曲线图

六、 问题一模型的建立与求解

6.1 数据可视化与分布规律描述

6.1.1 品类之间的分布规律

生鲜商超共有 6 大品类蔬菜，统计时间跨度为 3 年（2020 年 7 月 1 日起至 2023 年 60 月 30 日截至），共计 1095 天数据。由于以品类为单位进行分析，因此我们将品类下属单品销量(S_{SI}^{T})之和作为品类销量(S_{CN}^{T}) ,由此构建了基于时间为自变量的各品类蔬菜日销量曲线图。之后将每一种品类按照由于篇幅所限，本文选出两种有代表性的曲线图详细描述，其余图见附录二。

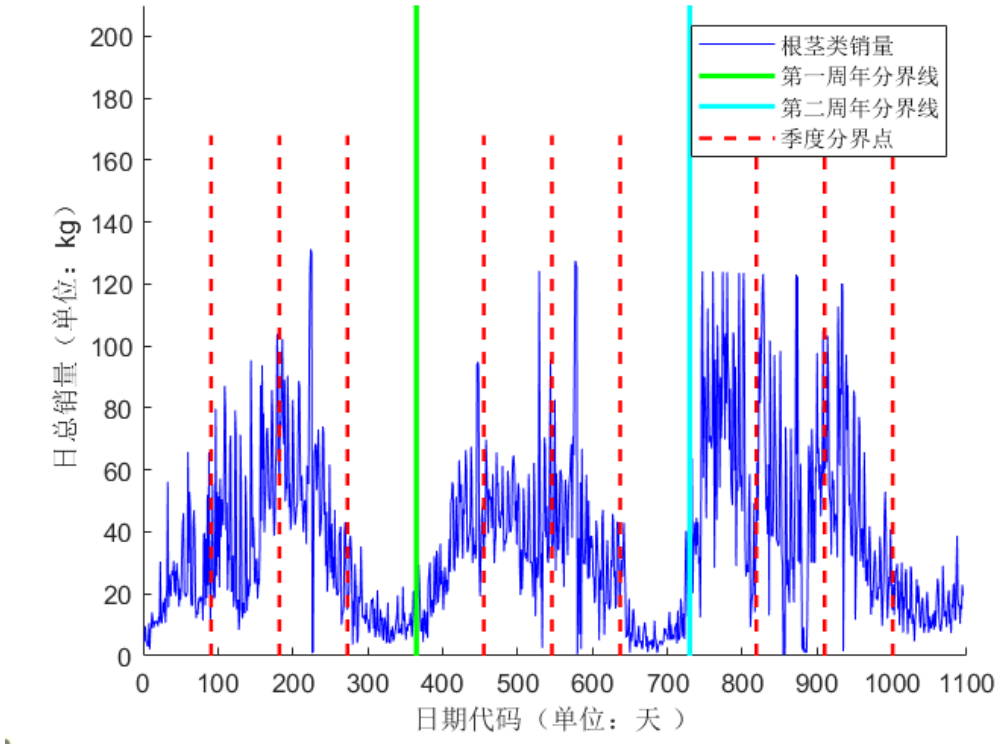


图 6.1 根茎类销量曲线图

由图 6.1 所展示的根茎类蔬菜销量曲线图可见，根茎类蔬菜销量具有明显的年度周期性，年度波动和年内波动程度均较大，规律性较强。具体到 1 年(1 个周期)之中，可以看到根茎类销量呈明显的先上升后下降趋势，在每年的 7 月份是销量最低的时间段，每天仅有不到 20kg 的销量，此后以较快的速度增长在第二年的 2、3 月份达到顶峰，日销量几乎可达 80kg 上下。可见按照季节变化合理地安排补货十分重要。

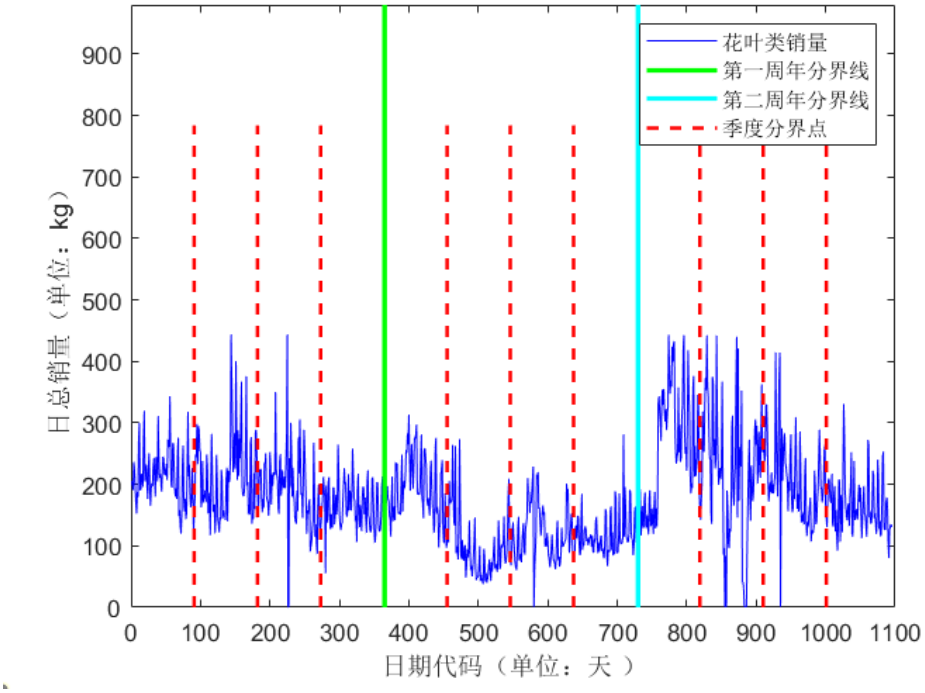


图 6.2 花叶类销量曲线图

由图 6.2 所展示的花叶类蔬菜销量曲线图可见，花叶类蔬菜的销量不具有明显的年度周期特性，而在一年之内，其销量基本趋于稳定状态，波动程度相对较小，在大部分时间段里日总销量均位于 200kg 附近，销量很大。结合花叶类蔬菜的下属单品种类我们推测，由于花叶蔬菜的市场需求量较大且较为稳定，一年四季均盛产相应的花叶类蔬菜，而花叶类蔬菜又作为最为常见的家用食材，自然经久不衰，呈现稳定特性。

表格 6.1 各品类蔬菜销量随时间分布规律总结

蔬菜品类	波动程度	周期规律	日总销量	预测风险
茄子类	小	强	少	低
食用菌类	中	较强	中	中
根茎类	小	强	少	中
花菜类	较小	弱	少	高
花叶类	大	弱	多	高
辣椒类	中	较弱	中	较低

6.1.2 单品之间的分布规律

生鲜商超共有 251 个单品蔬菜记录，统计时间跨度为 3 年（2020 年 7 月 1 日起至 2023 年 6 月 30 日截至），共计 1095 天数据。由于本问关注单品之间的销量关系，本文用单品日销量S_{CN}^{T}为纵坐标，天数为横坐标由此构建了基于时间为自变量的各单品蔬菜日销量曲线图，考虑到单品种类及数量的复杂性以及文章篇幅所限，本文主要截取了几类销量较高的单品进行统计，对于其余销量少的单品不予考虑。

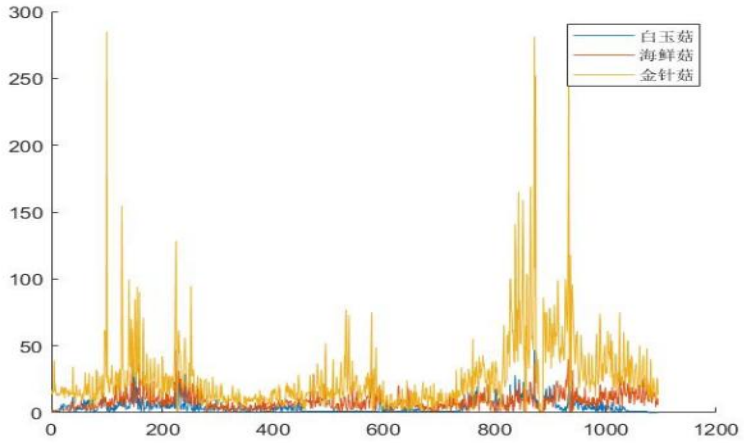


图 6.3 菌类下属单品销量曲线图

从同一类下的单品销量明显可以看出，同一类下的单品销量存在“趋势相同、销量不同”的特点，可以看到各类别之间的销量差距较大，在食用菌类中，金针菇、海鲜菇仅两种单品对该品类的总销量贡献度便达到了 95%，这提示我们在描述单品之间销量分布规律时可以采用“代表制”即每一类蔬菜种选取其中销量最高一种单品的来代表其他单品（该法仅限第一问的分布规律描述），这样使得 251 个单品简化成 6 个单品来描述。

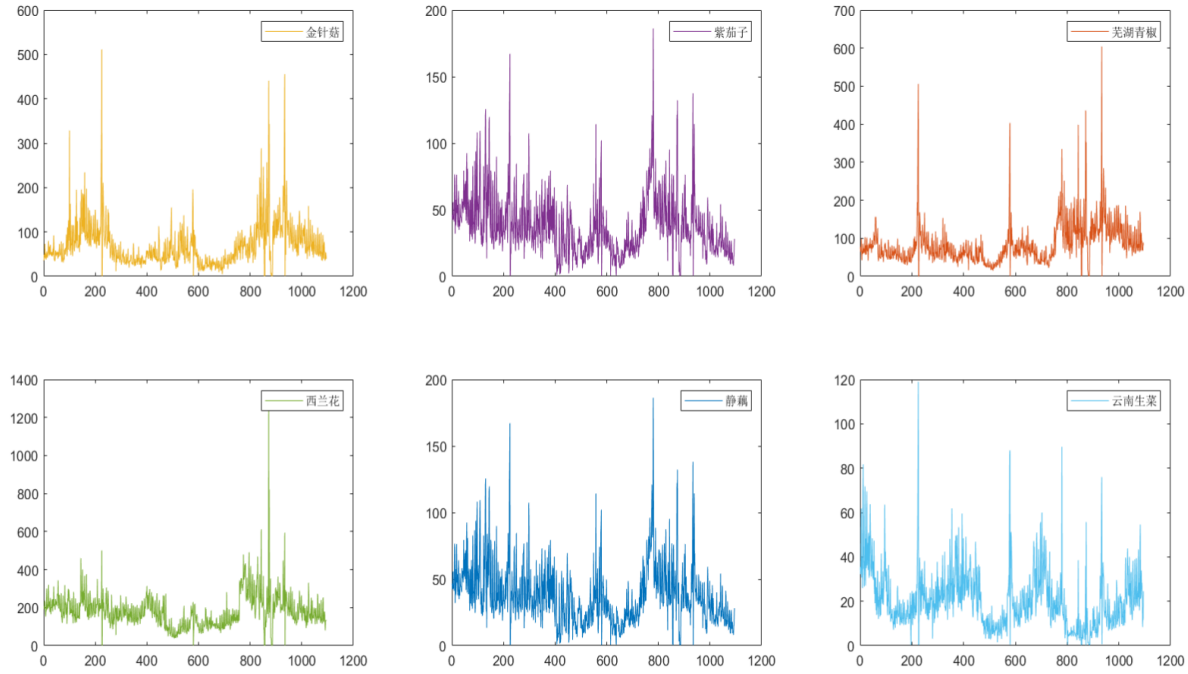


图 6.4 不同品类代表性单品销量曲线图

由图 6.4 可知，金针菇、云南生菜单品随时间具有明显的周期性，波动性也较为明显，应按照时令对补货策略及时及进行调整，而静藕、西兰花等单品的周期性不明显，且存在时间长度不等的断供期。

6.2 Spearman 相关性量化分析

在 6.1 中本文采用数据可视化结合描述的方法对两种对象的销量分布进行了描述，但仅靠肉眼观察所得结果的可靠性不高，定性分析难以对进货与定价策略做出精确的指导。在这里本文将采用 Spearman 相关系数进行相关性分析，对长达 3 年的 6 种蔬菜品类和销量排名前 30 的单品做相关性分析。计算流程如下：

6.2.1 数据特征条件分析

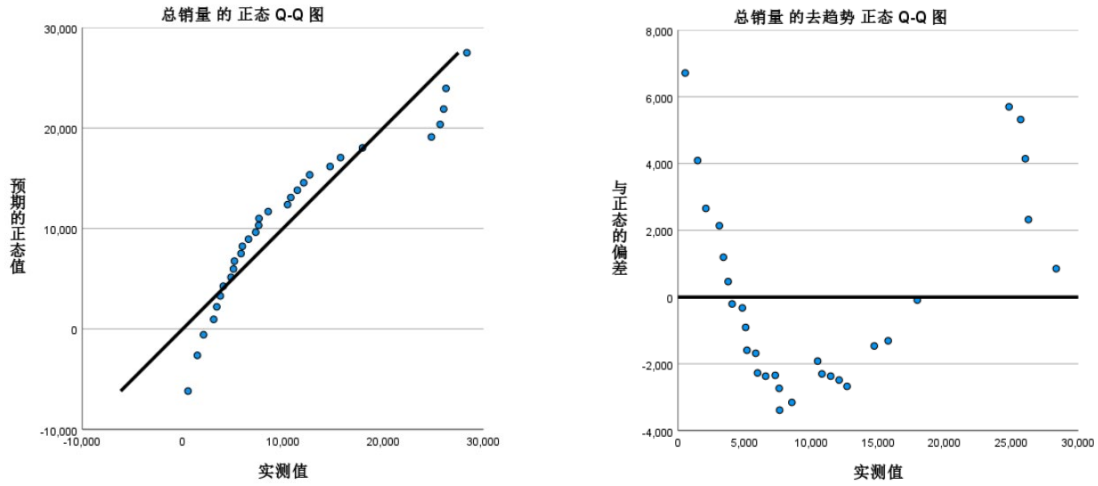


图 6.5 单品总销量的 Q-Q 图

在进行相关性检验之前，我们对各单品之间销量绘制了 Q-Q 图与矩阵散点图，发现其分布也不符合标准的正态分布。

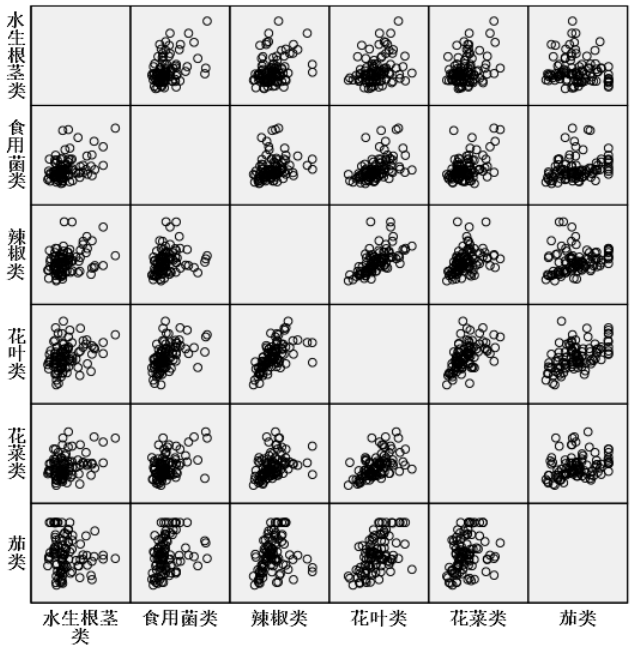


图 6.6 六大品类的矩阵散点图

我们利用 SPSS 绘制出品类销量之间的矩阵散点图，发现销量之间的线性关系不明显。以上两种结果均表明皮尔逊(Pearson)相关性检验、灰色关联度分析等方法不会得到较好的分析结果，经过综合分析对比后本文决定采用 Spearman 相关性检验。

6.2.2 模型原理与使用

斯皮尔曼（Spearman）相关性分析是一种常见的用于评估两个变量相关性强烈程度以及方向性关系的分析方法。其基本思想是利用两个变量的数值偏离程度来判断其联系程度和方向。通常运用此方法来分析各个因素对于结果的影响程度。但是普通的斯皮尔曼相关性分析无法体现随时间变化的维度，只可以分析出在某个时刻下各菜品销量之间的相关关系。因此为了将时间的动态变化过程融入其中，本文将 3 年的历史数据进行了分段处理，综合考量菜品的季节周期长度和数据处理的复杂程度，决定将 3 年划分为每年 4 个季度共计 12 各季度。再将每个季度的销量作为某一品类或单品的特征指标，记为 1 季度销售总量、2 季度销售总量…12 季度销售总量，用 $Q_i^S(i=1,2,...,12)$表示。如此每一品类便具有 12 个特征指标，巧妙地将时间动态变化的维度予以考量，增加了模型的客观性与精确度。

方法实现上我们用 SPSS 软件进行计算，设置参数如下：

指标个数	指标权重	无量纲处理方式	分辨系数ρ
12	均为 1/12	归一化	0.5

6.2.3 相关性分析结果展示

经过 SPSS 软件 Spearman 相关性分析并将数据整理后得到如下表格：

表 6.2 6 种蔬菜品类之间的相关系数矩阵

蔬菜品类	根茎类	辣椒类	花菜类	茄类	花叶类	食用菌
根茎类	1	0.434	0.601	-0.769	0.566	0.685
辣椒类	0.434	1	0.427	-0.392	0.552	0.678
花菜类	0.601	0.427	1	-0.189	0.860	0.434
茄类	-0.769	-0.392	-0.189	1	-0.217	-0.650
花叶类	0.566	0.552	0.860	-0.217	1	0.522
食用菌	0.685	0.678	0.434	-0.650	0.552	1

由于表格的等效对称性，只取左下半部分数据进行分析，可见相关性最强的蔬菜品类是花叶类与花菜类，二者相关系数高达 0.86，其次是茄类与食用菌的相关性达到 -0.769 呈明显的负相关，即可认为在食用菌销售旺季时茄子正处于销售淡季。另外还有花菜类与根茎类、食用菌与根茎类均超过了 0.6，三者往往一起售卖。茄子类与其他品类均呈不同程度的负相关。

同理经过筛选我们选取 251 个单品种销量排名前 30 的单品，分别是金针菇、西陕香菇、海鲜菇、紫茄子、青茄子、云南生菜等共计 30 个单品，同样时间分为 12 个季度指标，构建 Spearman 相关性分析。由于篇幅所限，30 个单品之间的相关系数矩

阵详见附录二，30 个单品之间的相关系数热力图如下图所示：

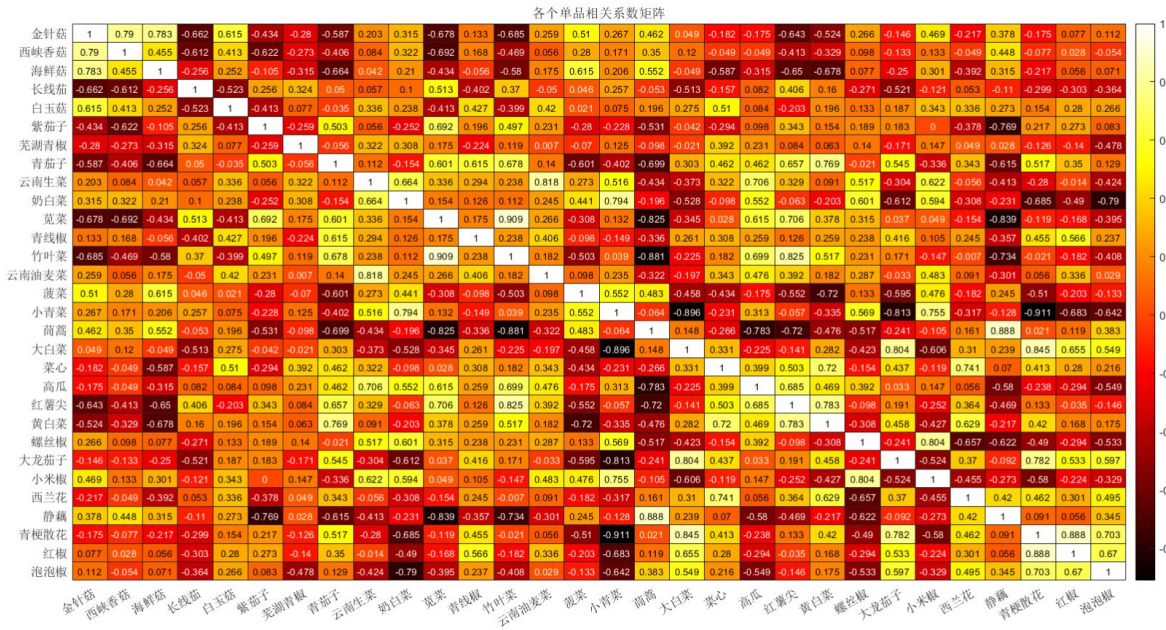


图 6.7　30 个单品之间相性热力图

由上图 6.2.2 可知金针菇与西陕香菇、小米椒之间相关性较强，可放在一起售卖；以及黄白菜和大龙茄子、大白菜和红薯尖等等共计 15 种组合均达到了 0.6 以上的相关性，而红薯尖与西陕香菇杜纳后螺丝椒与芜湖青椒则具有-0.6 以下的明显负相关性，因此在考虑进货与售卖品类时，可以结合当地时令与品类相关性进行策略调整，会更有利于销售，避免蔬菜滞留产生浪费与经济损失。

七、 问题二模型的建立与求解

7.1 多项式拟合确定各蔬菜品类销量与售价的关系

基于假设，本问认为蔬菜品类的销量仅与品类定价有关之后用 MATLAB 的 cftool 工具对已知数据进行非线性拟合，可以得到不同品类销售定价与销售量的关系。

本题明确要求按照品类为单位，因此每个单品便可不予以区分，将当日（T 相同）该品类下所有单品销量S_{SI}求和作为品类总销量S_{CN}，即：

$$S_{CN}^{T}=\sum_{I=1}^{251}S_{SI}^{T} \tag{5}$$

其中$N\in(1,2,...,6)$代表品类，n_N代表N品类下包含的所有单品总数，I代表单品。本文采用单品定价加权平均的方式作为品类定价的方法，即：

$$P_{CN}^{T}=\frac{\omega_1\cdot P_{S1}+\omega_2\cdot P_{S2}+....+\omega_n\cdot P_{SI}}{\sum_{I=1}^{n_I} m_{SI}} \tag{6}$$

将P_{CN}作为自变量，S_{CN}作为因变量，n_I是 I 类单品种类总数绘制六种品类的P_{CN}-S_{CN}散点图， MATLAB 的 cftool 工具箱有许多拟合方法，这里为使模型简洁，采用较为常用的多项式拟合方法。拟合函数如图 7.2 所示，拟合详细结果如表 7.3 所示：

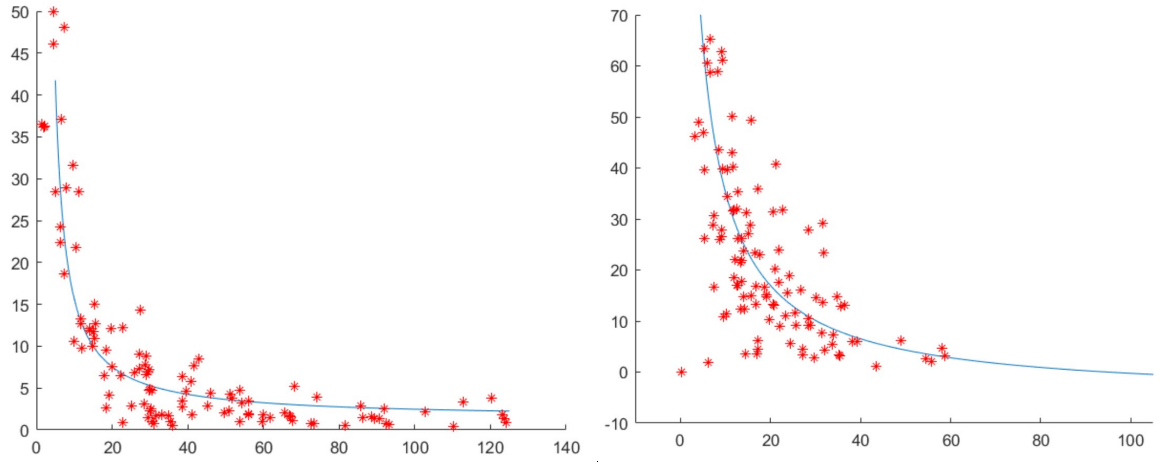


图 7.1 举例茄类（左）与根茎类（右）销量定价关系的拟合函数效果图

表 7.1 六种品类不同定价与销售量之间的关系

$f(x)=(p_1\cdot x^2+p_2\cdot x+p_3)/(x^2+q_1\cdot x+q_2)$ (p_1,p_2,q_1,q_2均为相关系数)						
不同蔬菜品类	茄类	花菜类	花叶类	辣椒类	食用菌	根茎类
p_1	1.37	-4.009	7.92	1.116	2.256	-4.988
p_2	103	1447	-927.9	423.8	835	476.7
p_3	-156.5	-1735	35760	-13.61	-341.4	-305.4
q_3	-3.975	149.7	-104.3	-23.42	23.51	1.389
q_4	4.276	-554.3	3153	2702	-18.46	-1.679
Adjusted R-square	0.962	0.4388	0.4734	0.2335	0.9221	0.5232

六种销量定价曲线的拟合程度高低不一，茄类、食用菌拟合优度较好，Adjusted R-square 均可以达到 0.9 以上，另外三类的效果则差强人意。在查阅相关文献后发现，在同一环境下卖同类产品（此处有理由认为所有品类均属于蔬菜）时，可以认为销量与售价具有固定的对应关系，因此本问选用拟合效果最好的函数作为品类售价与销量的关系函数，可见二者之间呈现明显的反比例函数性质，符合基本市场规律。

7.2 基于混沌时间序列预测模型

7.2.1 模型原理

混沌时间序列预测，是一种新型的、针对短期的非线性系统预测理论，其研究目标为如何通过时间序列通过相空间重构，从另一个维度和视角来辨识系统，寻找内藏规律，预测未来的走势。 由图可知，本文截取的 30 天样本的补货价，且与时间成非线性关系，其规律较难把握，具有一定的混沌性，因此考虑用混沌时间序列预测模型进行预测。

7.2.2 模型建立步骤

针对补货成本价混沌时间序列$\{X_t\}$，$t=1$，2，.....，n，利用相空间重构技术和$Takens$定理，嵌入维度为d，时滞为τ，重构相空间中的矢量点可表示为：

$$X_t=(x_t,x_{t+\tau},...,x_{t+(d-1)\tau})^T \tag{7}$$

根据非线性动力学，建立混沌时间序列预测模型，可以对某一品类蔬菜的进货成本价进行混沌局部线性多步预测。设置预测步长为d，根据$Takens$定理，建立混沌时间局部线性预测函数，即：

$$X_{t+1}=aX_t+bI$$

其中，a 和 b 是拟合系数，I 是各分量均为 1 的 d 维列向量。对于价格预测的混沌时间序列的，在重构相空间的矢量点中，选取一定的领域半径ε，找出临近X_P的k个矢量点，X_{P1}，X_{P2}，.....，X_{Pk}，使其满足：

$$|X_{Pj}-X_P|\leqslant\varepsilon \quad (j=1，2，.....，k)$$

定义ξ_i和ξ_p和k个矢量点的一步迭代点X_{P1+1}，X_{P2+1}，.....，X_{Pk+1}，如下定义：

$$P=\frac{1}{k}\sum_{i=1}^{k}X_{Pi}，\ \xi_{Pi}=X_{Pi}-P，\ \xi_P=X_P-P$$

$$\tilde{P}=\frac{1}{k}\sum_{i=1}^{k}X_{Pi+1}，\ \tilde{\xi}_i=X_{Pi+1}-\tilde{P}，\ \tilde{\xi}_P=X_P-\tilde{P}$$

最小二乘法拟合出 a 与 b 的拟合系数，即最小化误差：

$$\min E=\min\frac{1}{k}\sum_{i=1}^{k}\left|\tilde{\xi}_i-a\xi_i-bI\right|^2$$

令误差 E 对拟合系数的偏导数为 0，即满足：

$$\frac{\partial E}{\partial a}=0，\ \frac{\partial E}{\partial b}=0$$

然后根据以下公式可以计算出矢量$\tilde{\xi}_P$，从而可以求出X_{P+1}：

$$\tilde{\xi}_P = a\xi_P + bI$$

之后同理可求出X_{P+T}。

7.2.3 预测的参数

在本文建立的混沌局部线性预测中，对于成本预测的参数设置如下：

表 7.2. 预测参数

步长T	滞后时间τ	嵌入维数d	领域半径ε
7	10	6	0.35

7.2.4 预测的结果

以根茎类预测结果为例，其中 30 天之前可以看到预测值与实际值的偏差，30 天之后则是本文预测的 7 天成本，图如下：

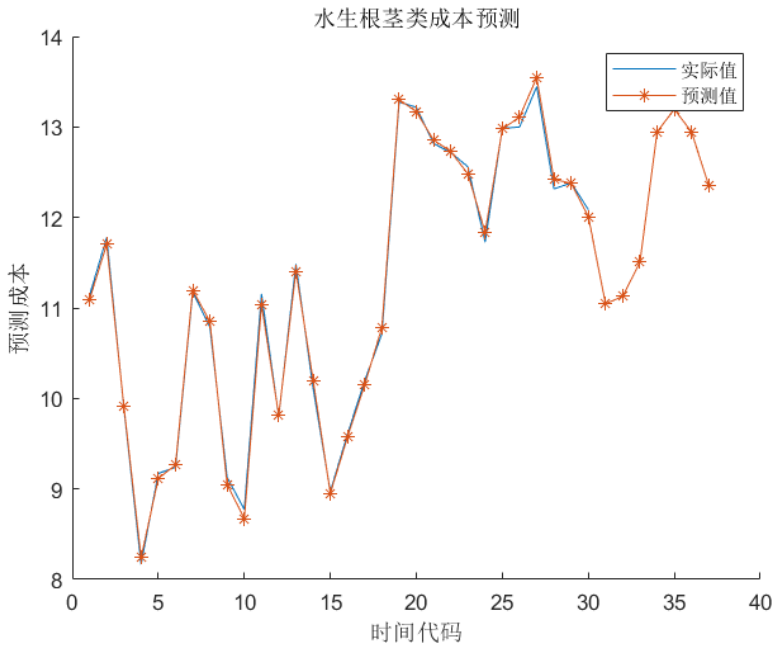


图 7.2 根茎类成本预测效果图

表 7.3 预测 6 种蔬菜品类未来七天成本（单位：元/千克）

日期	茄类	花菜类	花叶类	辣椒类	食用菌	根茎类
1	5.1	8.15	3.9	5.11	4.88	11.05
2	4.29	8.48	3.94	5.78	4.91	11.13
3	5.71	8.25	3.49	5.39	4.15	11.51
4	5.24	8.37	3.49	5.24	3.12	12.95
5	4.7	8.48	3.34	5.4	3.47	13.2
6	5.03	8.83	3.9	5.1	3.71	12.94
7	4.8	8.1	3.37	5.13	4.64	12.35

7.3 基于非线性规划的定价成本预测模型建立

本题要求商超的利润取得最大值时的补货量与定价策略。首先建立以利润为函数值的目标函数：

$$W=\sum_{T=1}^{7}\left(\sum_{N=1}^{6}f(S_{CN}^{T})\times S_{CN}^{T}-\sum_{N=1}^{6}C_{CN}^{T}B_{CN}^{T}\times（1-L_{CN}）\right) \tag{8}$$

$$C_{CN}^{T}=S_{CN}^{T}\times(1+L_{CN}) \tag{9}$$

其中，S_{CN}^{T}为N品类蔬菜T天的实际销量（即已减去退货后的销量），C_{CN}^{T}为N品类蔬菜的T日补货量，L_{CN}为N品类蔬菜的损失量，其中包括运输损失和未卖完损失。对于函数变量范围做必要约束如下：

1）每种品类补货量范围限制

$$C_{N}^{MAX}\leqslant C_{N}\leqslant C_{N}^{MAX} \tag{10}$$

2）每种品类进货量范围限制

$$S_{CN}^{MIN}\leqslant S_{CN}\leqslant S_{CN}^{MAX} \tag{11}$$

3）每种品类蔬菜定价范围限制

$$P_{CN}^{MIN}\leqslant P_{CN}\leqslant P_{CN}^{MAX} \tag{12}$$

4)统计6种品类蔬菜的平均损失量（本小问为简化模型，损失量按照常量即历史数据取加权平均值损失率），统计结果如下表所示：

表7.4 六种品类蔬菜各自损率

茄类	花菜类	花叶类	辣椒类	食用菌	根茎类
1.32%	10.98%	14.16%	8.47%	7.84%	5.75%

5)经过数据统计分析后确定6种品类蔬菜中的定价、补货量、进货量上下限如下表所示：

表7.5 补货量、销量、定价范围限制

蔬菜品类	C_{N}^{MIN}	C_{N}^{MAX}	S_{CN}^{MIN}	S_{CN}^{MAX}	P_{CN}^{MIN}	P_{CN}^{MAX}
茄类	4.05	5.68	8.41	54.5	8.65	10.5
花菜类	7.84	10.33	10.17	33.24	7.5	15.3

花叶类	2.96	3.68	81.52	253.45	6.35	8.36
辣椒类	2.66	4.14	57.35	169.2	4.67	7.98
食用菌	2.88	4.51	34.9	98.84	8.52	11.53
根茎类	8.16	13.45	6.83	38.77	11.42	17.85

经过化简易得该利润函数可以化简为补货量为自变量二次函数，使用 MATLAB 软件求解得到在七日产生总最大利润时 4362.19 元时，6 种品类的蔬菜的补货量与定价如下表所示：

表 7.6　7 月 1~7 日六种品类的蔬菜的建议补货量(单位：千克)

补货量（千克）	茄类	花菜类	花叶类	辣椒类	食用菌	根茎类
7 月 1 日	21.19	13.99	295	93	58.7	24.92
7 月 2 日	23.9	12.52	295	82.5	58.2	29.82
7 月 3 日	15.06	10.31	300	88.3	72.6	26.7
7 月 4 日	25	13.8	300	90.7	113.2	28.78
7 月 5 日	23.92	12.47	295	88	94.4	25.31
7 月 6 日	16.35	11.75	300	93.2	85.3	26.38
7 月 7 日	24.39	14.47	300	92.6	62.6	20.47

表 7.7　7 月 1~7 日六种品类的蔬菜的建议定价(单位：元/千克)

定价（元/千克）	茄类	花菜类	花叶类	辣椒类	食用菌	根茎类
7 月 1 日	7.23	10.69	7.55	5.55	12.3	13.97
7 月 2 日	6.49	11.22	7.55	5.78	12.36	11.06
7 月 3 日	10.08	12.4	7.56	5.66	10.88	12.8
7 月 4 日	6.24	10.75	7.56	5.6	8.36	11.6
7 月 5 日	6.48	11.24	7.55	5.66	9.31	13.7
7 月 6 日	9.27	11.56	7.56	5.55	9.89	13
7 月 7 日	6.37	10.54	7.56	5.56	11.86	17.7

八、 问题三模型的建立与求解

8.1 基于遗传算法求解补货与定价策略的模型

本题属于一个非线性优化问题，变量个数多，含有大量不等式约束条件，对于这种优化问题，一般的求解方法效率较低且难以达到全局最优的效果，而遗传算法（GA）具有良好的非线性寻优能力使得模型具有更好的预测效果。为此本文决定采用遗传算法求解该问题。

8.1.1 非线性规划模型建立

总收益应包含三个部分:成本加成定价产生正向收益；补货产生负向收益；打折销售产生正向收益；注意损耗产生的负向收益其实已经被补货成本所包含，不需额外计算。基于此构建出目标函数：

$$W=\sum_{I=1}^{Z}P_{SI}\times S_{SI}-\sum_{I=1}^{Z}C_{SI}B_{SI}+\sum_{I=1}^{Z}(B_{SI}-S_{SI})\times P_{SI}\times F_{SI} \tag{13}$$

此处与第二问不同的地方在于，本题还应通过打折的方式促进消费，根据问题二中拟合出的定价与销量关系函数，我们有理由认为，只要定价足够低，商超一定可以卖完当日的蔬菜，因此不考虑剩余损失问题。

其中W为商超在 7 月 1 日的总收益，Z 为销售单品数量，P_{SI}为单品 I 的成本加成定价，顾名思义本文构建了定价与成本的关系（17），其中m为加成率，如此操作也是为了便于实际中商超工作人员按照比率手册的定价。S_{SI}是单品 I 的未打折销量，C_{SI},B_{SI}分别是单品 I 在 7.1 日的补货成本与补货量，F为打折率，$B_{SI}-S_{SI}$为单品 I 的打折出售量。

另外题目还规定有其他约束，可售单品总数控制在 27-33 个，最小陈列量大于等于 2.5kg，如下：

$$\begin{cases}27\leqslant Z\leqslant 33\\ B_{SI}^{T}\geqslant 2.5\end{cases} \tag{14}$$

销售量不可能超过补货量，因此有如下约束：

$$P_{SI}\leqslant C_{SI}\times(1+m\%) \tag{15}$$

销售量与售价存在反比例函数的相互制约关系，即等式约束：

$$f(P_{SI})=S_{SI} \tag{16}$$

同时也将第二问的上下限约束不等式作为本题的约束条件。另外利用第二问混沌

时间序列预测模型预测出 7.1 日的各单品进货成本，详见附录二。综上所述，整理可以得如下规划公式：

$$MAX \quad W=\sum_{I=1}^{Z} P_{SI}\times S_{SI}-\sum_{I=1}^{Z} C_{SI}B_{SI}+\sum_{I=1}^{Z}(B_{SI}-S_{SI})\times P_{SI}\times F_{SI}$$

$$s.t.\begin{cases} C_{SI}^{MAX}\leqslant C_{SI}\leqslant C_{SI}^{MAX} \\ P_{SI}^{MAX}\leqslant P_{SI}\leqslant P_{SI}^{MAX} \\ S_{SI}^{MAX}\leqslant S_{SI}\leqslant S_{SI}^{MAX} \\ P_{SI}\leqslant C_{SI}\times(1+m\%) \\ f(P_{SI})=S_{SI} \\ B_{SI}\geqslant 2.5 \\ 27\leqslant Z\leqslant 33 \\ S_{SI},C_{SI},P_{SI}>0 \end{cases}$$

8.1.2 遗传算法原理

遗传算法（Genetic Algorithm）是一种借鉴生物进化论及其演化规律，用优胜劣汰作为其基本思想的全局概率寻优算法。其本质是对结构对象进行抽象，不存在求导与连续性的函数限制。导向式概率化的寻优法决定了其具有更好的全局寻优能力。同时该算法也可以自动获取和指导优化的搜索空间，自适应高速搜索方向，不需要确定规则，遗传算法在现代计算机算法中占有很高的地位，被广泛应用于组合优化、机器学习、信号处理等领域。

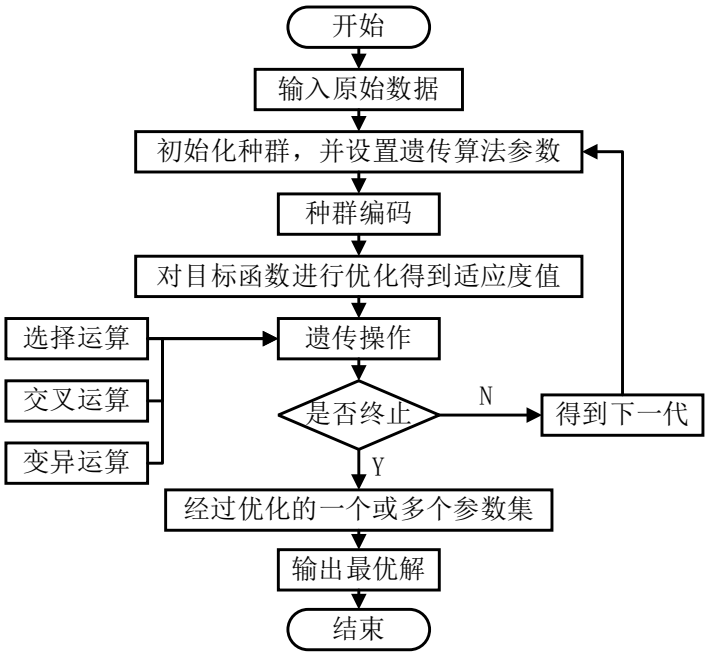


图 8.1 遗传算法求解流程图

8.1.3 遗传算法求解规划模型步骤

经过对 2023 年 6 月 24~30 日的可售品种的统计，最终共计找到 49 个可售品种，接着利用上一问的算法预测出这 49 种单品在 7 月 1 日的补货价格记为 C_{SI}^{T}，$I \in (1,...,49)$，详细见附录二。

1）编码策略选择

为使编码、解码操作简单，本文采用二进制编码方式，运算精度定为小数点后 4 位，对一个变量的二进位数串数用以下公式计算：

$$2^{k-1} \leqslant (U-L) * 10^{4} \leqslant 2^{k}$$

其中，k 表示二进位数长度。本次算法中，选取$L=0$和$U=20$，得到$k=11$，即用长度为 11 的二进制数表示问题的解。

2）种群初始化：

实际操作中，可根据情况将群体规模设置为 n，每个个体都通过$L+random(\ )*(U-L)$这个随机函数在 L 至 U 中随机选择 n 个数据，其中 U 为最大值，L 为最小值。在本算法中，设置成本加成定价和补货量作为待求变量，其余参数设置如下表：

种群数量	种群迭代次数	交叉概率	变异概率
200	300	0.9	0.005

3）确定适应度函数

本题的适应度函数为生鲜商超在 7 月 1 日获得的总收益。适应度函数即为规划模型中的目标函数，如下：

$$W=\sum_{I=1}^{Z}P_{SI}\times S_{SI}-\sum_{I=1}^{Z}C_{SI}B_{SI}+\sum_{I=1}^{Z}(B_{SI}-S_{SI})\times P_{SI}\times F_{SI} \tag{17}$$

4）选择操作

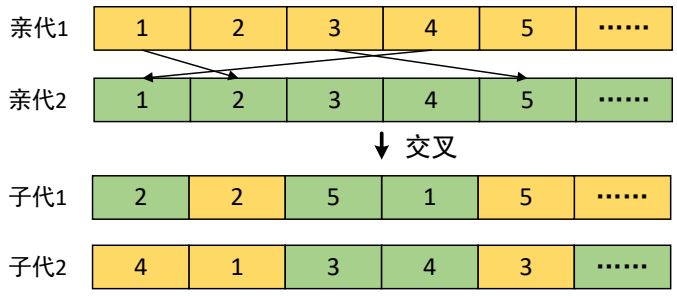
本文选用适应度比例法，公式如下：

$$f_i=k/F_i\ ,\ P_i=\frac{f_i}{\Pi}\ ,\ \Pi=\sum_{j=1}^{N}f_j \tag{18}$$

其中，个体i的适应度值为F_i，k为系数，N为种群个体的总数。

5）交叉操作

在第一问中的求解可知，不同种单品之间的销量之间存在或大或小的关系，因此补货量之间可能有比较高的相关性，因此在遗传算法中的交叉操作也需要考虑到不同单品之间的相关性，交叉操作如下：



交叉位点1	交叉位点2	相关性
1	2	0. 95
3	5	0. 88
4	1	0. 66
……	……	……

（表内虚拟数据仅供解释参考）

图 8.2 遗传算法交叉示意图

依据初始化时设定的交叉率，选取某个染色体，并列出所有的交叉组合，依据两种小类之间的相关性进行排序，设置染色体交叉覆盖率为 0.7~1，由上述排列选取前70%~100%的组合进行编译选取第i个个体的第j个两个基因a_{ij}进行变异操作，公式为：

$$a_{ij}=\begin{cases}a_{ij}+(a_{ij}-a_{\min})f(g) & r\geqslant 0.5\\ a_{ij}+(a_{\max}-a_{ij})f(g) & r<0.5\end{cases} \tag{19}$$

其中，b是$[0,1]$之间的任意数。

6）变异操作

选取第i个个体的第j个基因a_{ij}进行变异操作，公式为：

$$a_{ij}=\begin{cases} a_{ij}+(a_{ij}-a_{\min})f(g) & r\geqslant 0.5 \\ a_{ij}+(a_{\max}-a_{ij})f(g) & r<0.5 \end{cases} \tag{20}$$

其中，基因a_{ij}的上限是$a_{\max}$，基因a_{ij}的下限是$a_{\min}$，$f(g)=r_2(10g/G_{\max})$，r_2是一个任意数，g 是当前的迭代次数，$G_{\max}$是最大进化次数，r是$[0,1]$之间的任意数。且由于补货量长期在一定的范围之内波动，所以上式的基因上限和下限可以由历史数据的最小值和最大值来替代。

7）优化模型求解结果

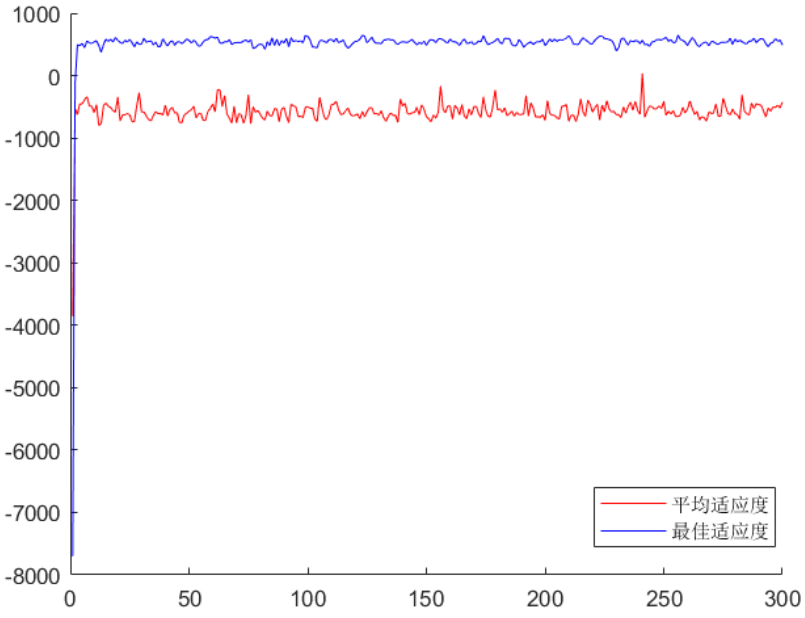


图 8.3 最优及平均个体最优适应度变化图

从图中可以看出，使用遗传算法求解效果收敛速度很快，收敛性较好，最佳适应度即当日最大盈利在 4000 元上下波动。

8）可售单品建议补货量

表 8.1 33 种单品 7.1 日补货量一栏表（单位：千克）

单品名称	长线茄	西兰花	菠菜	…	西山花菇	红莲藕带
补货量	5.88	14.97	2.65	…	6.19	3.18

9）可售单品建议成本加成定价

表 8.2 可售单品成本加成定价策略

单品名称	补货成本	加成率	定价	打折
长线茄	6.98	79%	12.5	90%
西兰花	7.6	65.7%	12.6	100%
菠菜	9.64	48.3%	14.3	88.2%
…			…	
西山花菇	15.6	60.2%	25	100%
红莲藕带	5.36	69.8%	9.1	78.6%

10）经策略优化后商超日收益额度对比统计结果

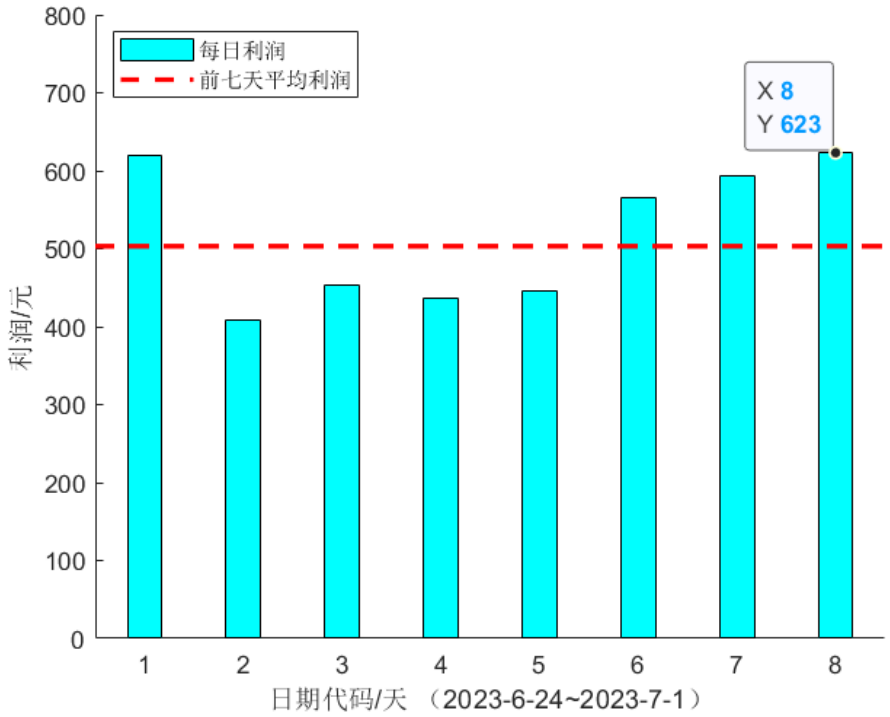


图 8.4 该商超公司 6.1-7.1 日每日盈利收入曲线图

由图可知，该商超公司在为采用定价策略优化时前三十天的收益处于可接受正常波动范围内，收益基本处于 400~600 元之间。而采用新的补货、定价、打折促销策略后，本文预计可在 7.1 日达到 623.02 元的总收益，比前七天平均收益相比涨幅为 19.7%。

九、 问题四模型的建立与求解

本题第二三问均是站在假设所有蔬菜的保质期只有一天的情况下家里的数学模型，然而生活常识告诉我们，并不是所有的生鲜蔬菜保质期只有一天。比如，西兰花

保质期大概有 3~5 天，金针菇常温保质期 1~2 天，冷藏 5 天。冷藏的保存虽然有助于延长保质期，但也需要投入更高的保险成本。而对于天生保质期较长的蔬菜，生鲜商超完全可以根据实际情况调整进货频率，一次性大量进货也有利于进货成本更加优惠。另外，关注天气情况的变化并基于此建立天气补偿系数也会增加模型的稳健性，基于此本文结合其他研究成果提出了定价与库存联合决策模型，可以作为模型后续推广的参考思路。

9.1 蔬菜定价与库存联合决策模型

9.1.1 建立变质库存与时间的微分方程

变质品的库存模型来源于 Ghare 和 Schrader 在总结传统经济批量模型（EOQ）上，首次构建了变质品的变质率为常数的变质库存模型，定义变质品在 t 时刻的库存变化。

$$\frac{dI(t)}{dt} = -D(t) - \lambda I(t) \tag{21}$$

其中，λ为产品的变质率，$I(t)$变质品在时刻 t 的库存水平，$D(t)$为易变质品 t 时刻需求率

根据变质库存模型，需求率和变质率是影响模型的主要因素。随后在针对变质库存的研究中，考虑产品变质速率、市场需求率、是否允许缺货、是否保鲜等关键因素，结合企业实际情况，考虑各种因素的相互搭配，构建出不同的库存模型。

9.1.2 蔬菜商品定价与库存联合决策模型

根据变质库存理论原理，在保鲜期和变质期的库存模型分别为

$$\frac{dI(t)}{dt} = -D(t), \quad 0 < t < 1 \tag{22}$$

$$\frac{dI_2(t)}{dt} = -\bar{D} - \lambda I_2(t) \ , (t_1 < t < T) \tag{23}$$

根据临界条件$I_2(t) = 0$, $I_1(t) = I_2(t)$，可得保鲜期与变质期蔬菜单品在 t 时间的库存水平。为此，求得商超对各蔬菜单品的订货量为$Q = I_1(0)$。据此构建利润模型:

$$\pi(p,t,T) = p\left[\int_0^{t_1} D\,dt + \int_{t_1}^{T} \bar{D}dt\right] - CQ \tag{24}$$

根据此方程得到最优解 p，将其代入得最优订货量 Q，可帮助商超更好地制定蔬菜商品的补货和定价决策。

十、 模型的分析与检验

问题 1 采用了 Spearman 相关性分析，在数据不符合正态分布、线性关系不明显的情况下，求得品类、单品之间销量的相关系数矩证，考虑到第一问中求出的相关性在后续模型中要继续使用，本文使用聚类分析进行交叉验证。在问题 1 各类蔬菜、各单品蔬菜销量已知的情况下，我们转而利用层次聚类，将蔬菜各品类、单品及其销量作为指标，进行聚类分析。首先进行数据处理使得变量统一，通过聚类得到分组，同组数据相关性较强，不同组数据相关性较弱。以此验证 Spearman 相关性分析结果的可靠性。

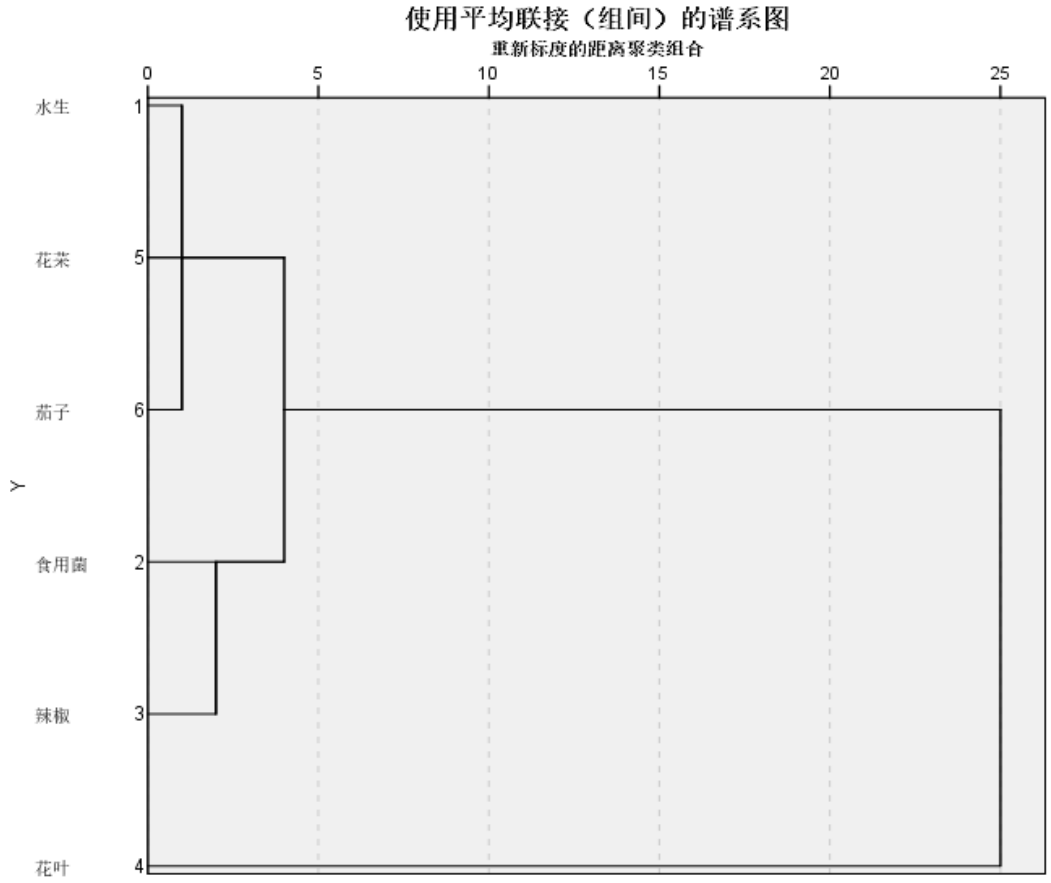


图 10.1　6 大品类聚类结果

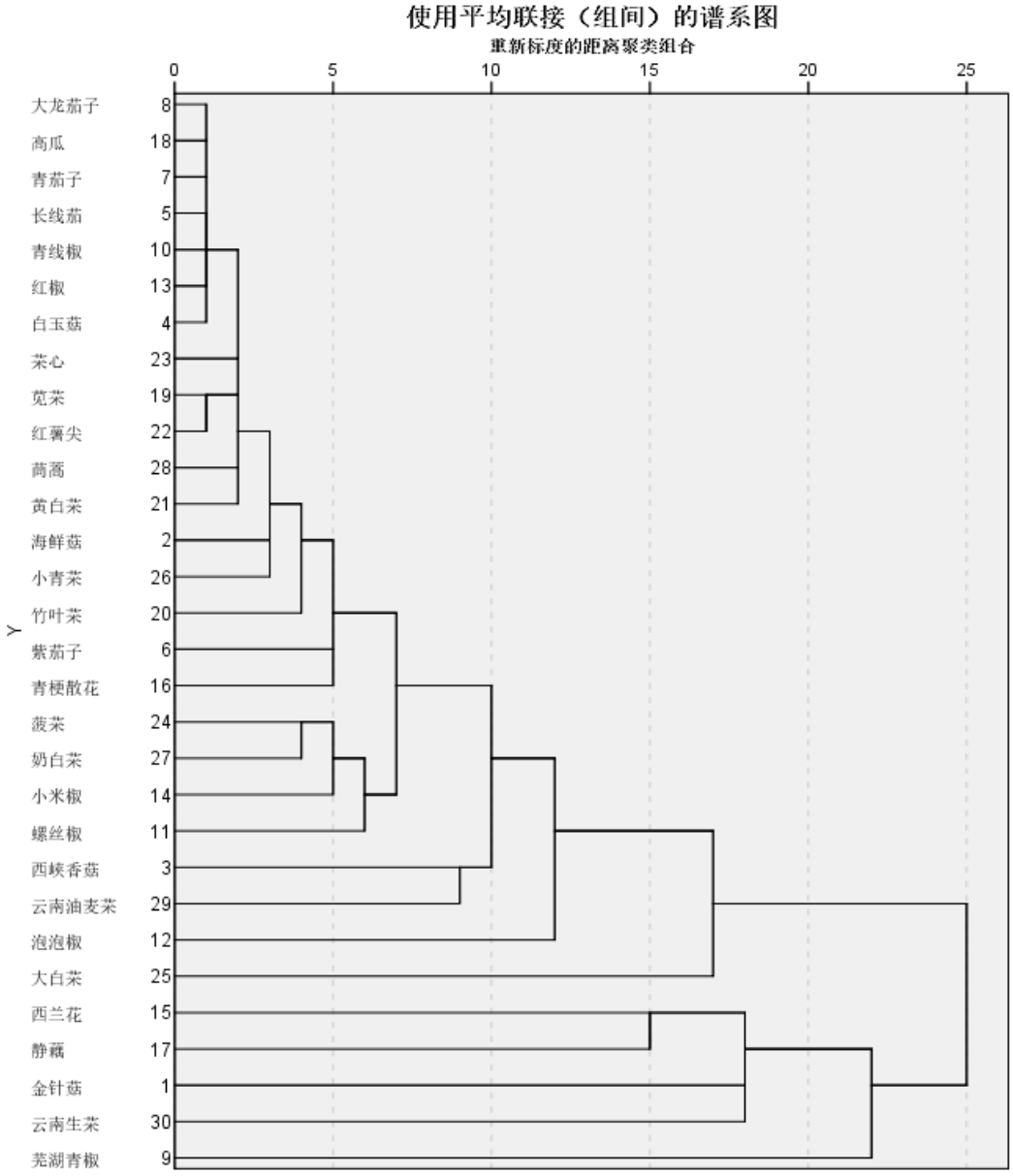


图 10.2　30 种单品聚类结果

将各蔬菜单品划分三类为例，即芜湖青椒为一类，静藕、金针菇、云南生菜为一类，其他蔬菜单品为一类，同类单品销售量相关性较强，不同类单品销售量相关性较弱。例如。红薯尖与竹叶菜在以上聚类分析中被划分为同类别，在 Spearman 相关性分析中，相关系数更是高达 0.825: 芜湖青椒在聚类模型中被单独划分，观察 Spearman 所求相关系数矩证可看出其销量与其他单品销量相关性较弱。两种模型所得结果基本一致，验证了 Spearman 相关性分析方法的可靠性。

十一、 模型的总结与评价

11.1 模型的优点

Spearman 相关性分析在第一问分析各品类/单品销量的关系等非线性特征指标时，采用将时间分段法求出每个时段的销量，这将原本单一维度的销量指划分成了具有时间特征的 12 个销量指标作为每个品类的补充指标，这巧妙地将 3 年的数据充分进行利用，这样将原本不具有时间动态的相关性分析融入了时间这一特征指标，使得相关性分析结果更加全面客观。

混沌时间序列预测模型能够很好的处理看似无序实则混沌的数据特征，本文用其预测补货成本取得了二次平滑指数预测、LSTM-ARIMA 时间序列预测均难以企及的效果，对于短期的进货成本预测具有深远的指导意义。

非线性回归模型传统方法难以计算出全局最优解，本文采用遗传算法进行计算。在增加了销售空间与品类的限制后，将收益函数作为适应度函数，结合单品之间的相关性定义交叉形式与变异概率和优先级，采用并行计算来提高运算速度，拥有较强的可并行性，利于得到商超收益最大时的补货量与定价策略。

11.2 模型的缺点

本文为减少计算量与模型复杂程度决定将 251 个单品做简化处理，只保留了销量较高的前 30 个作为问题一相关性分析的对象，这影响了模型的客观性和全面性。

虽然遗传算法解规划模型具有较快的速度与较强的全局最优搜索能力。但是，遗传算法复杂的操作过程，使网络的搜索时间随着研究问题的复杂程度不同呈现指数倍增长。遗传算法在得到最优解时收敛缓慢甚至停止收敛。

十二、 参考文献

[1] 杜荣.产品定价与促销中的决策优化问题研究[D].西安电子科技大学.2004 年 4 月.
[2] 谢虎，杨占杰，张伟，何超林，谢型浪，潘睿．基于改进遗传算法的主动配电网优化定价策略研究[J].2022 年.电测与仪表.3-4.
[3] 康莎.不同需求特征下生鲜农产品双渠道销售定价与库存补货联合决策研究[D].重庆交通大学,2022 年 6 月 13 日.
[4] 杨天山，袁功林．考虑绿色配送技术投资的易腐品动态定价与补货策略研究[J].中国管理科学. https://doi.org/10.16381/j.cnki.issn1003-207x.2021.1016.
[5] 辛侠云.易逝生鲜产品多阶段定价研究[D].重庆大学.2022 年 6 月.
[6] 卢亚杰.我国超市优质生鲜蔬菜动态定价问题研究[D].北京交通大学.2010 年 6 月.
[7] 张春明，张彤，王少军.基于遗传算法的逆向供应链 Stackelberg 主从定价策略分析.价值工程 2006 年第 10 期.49-50.
[8] 吴祖荣.市场营销中的产品定价优化决策研究[D].西安电子科技大学.2009 年 1 月.
[9] 赵佳鑫，高岳林，陈群林.一种求解非线性规划问题的粒子群算法[J].宁夏大学学报（自然科学版）第 38 卷第 1 期.15-16.
[10] 刘佳.二次指数平滑预测模型的一点探讨[J].科技视界.155-156.
[11] 张婧.基于数据挖掘的零售业商品销售预测研究[D].四川师范大学.2008 年 4 月.

附录

附录 1

介绍：支撑材料的文件列表

1. solve1.m

```
%% 求解第二问模型
clc,clear;
%导入数据
load predict.mat;
c=predict;
rate=[1.32204725500000, 10.9839178300000, 14.1584886500000 8.47212605900000, 7.84135120100000, 5.74567672500000]/100;
p1= [1.357, -4.009, 7.92, 1.116, 2.256, -4.988];
p2= [103, 1447, -927.9, 423.8, 835, 476.7];
p3=[-156.5, -1735, 35760, -13.61, -341.4, -305.4];
q1=[-3.975, 149.7, -104.3, -23.42, 23.51, 1.389];
q2=[4.276, -554.3, 3153, 2702, -18.46, -1.679];
x=zeros(7,6);
f=zeros(7,6);
p=zeros(7,6);
%由于此处目标函数形式较为简单，可以通过暴力搜索算法进行求解
for i=1:7
    for j=1:6
        x_best=0;
        f_best=-1000000;
        xline=0:0.1:300;
        for k=1:length(xline)
            xin=xline(k);
            xiao=xin*(1-rate(j));
            fs=(p1(j)*xiao^2+p2(j)*xiao+p3(j))/(xiao^2+q1(j)*xiao+q2(j))*xiao-c(i,j)*xin;
            ps=(p1(j)*xiao^2+p2(j)*xiao+p3(j))/(xiao^2+q1(j)*xiao+q2(j))

            if fs>f_best
                f_best=fs;
                x_best=xin;
                p_best=ps;
            end
        end
        f(i,j)=f_best;
        x(i,j)=x_best;
        p(i,j)=p_best;
    end
end
```

```
%显示总利润
sum(f);
sum(ans);

heat_plot.m
%% 绘制热力图
%加载数据
load sale_heat.mat;
load label_heat.mat;
h=heatmap(label_heat,label_heat,sale_heat,'Title','各个单品相关系数矩阵
');
%调整配色
colormap('hot');

plot_bar.m
%% 绘制往日利润变化图和最后计算结果
%导入数据
y=[619.940779,408.4200207,453.3515,435.8932185,445.5370415,564.5875329,5
94,623.02];
%计算前七天利润的平均值
ymean=mean(y(1:7));
hold on
bar(y,0.35,'c')
line([0.5 8.5],[ymean ymean],'color','r','linewidth',2,'linestyle','--
');
hold off
%设置绘图参数
xlabel('日期代码/天 （2023-6-24~2023-7-1）');
ylabel('利润/元');
legend('每日利润','前七天平均利润');
axis([0.5 8.5 0 800]);

obj_fun6.m
function v=obj_fun6(x)
v=(-4.988*x.^2+476.7*x-305.4)./(x.^2+1.389*x-1.679);
end

obj_fun5.m
function v=obj_fun5(x)
v=(2.256*x.^2-835*x-341.4)./(x.^2+23.51*x-18.46);
end

obj_fun4.m
function v=obj_fun4(x)
```

```
v=(1.116*x.^2+423.8*x-13.61)./(x.^2-23.42*x+2702);
end

obj_fun3.m
function v=obj_fun3(x)
v=(7.92*x.^2-927.9*x+37560)./(x.^2-104.3*x+3153);
end

obj_fun2.m
function v=obj_fun2(x)
v=(-4.009*x.^2+1447*x-1735)./(x.^2+149.7*x-554.3);
end

obj_fun1.m
function v=obj_fun1(x)
v=(1.357*x.^2+103*x-156.5)./(x.^2-3.975*x+4.276);
end

ga_sel.m
function new_chrom=ga_sel(fitness,chrom,m)
        pro=fitness/sum(fitness);  %采用轮盘赌的方式进行选择
           pro_sum=zeros(m,1);
           index=[];
           for i=1:m
               pro_sum(i)=sum(pro(1:i));
           end
           for i=1:m
               p=rand(1);
               for j=1:m
                   if p<pro_sum(1)
                       index=[index,j];
                       break;
                   elseif p<pro_sum(j)&&p>pro_sum(j-1)
                       index=[index,j];
                   end
               end
           end
           new_chrom=chrom(index,:);
   end

   ga_mut.m
   function new_chrom=ga_mut(chrom,mu,m,narvs,x_up,x_dw)
   for i=1:m
       for j=1:narvs
```

```
123.            %低于设定变异率则进行交叉
124.            if rand(1)<mu
125.                chrom(i,j)=x_dw(j)+(x_up(j)-x_dw(j))*rand(1);
126.            end
127.        end
128.    end
129.    chrom=check(chrom,x_up,x_dw);
130.    new_chrom=chrom;
131.    end
132.
133.    ga_cro.m
134.    function new_chrom=ga_cro(chrom,cro,n,narvs,x_up,x_dw,relation)
135.    for i=1:n
136.            pcro=rand;
137.            if pcro<cro    %若随机数小于交叉率，进行交叉，否则不交叉
138.                x=chrom(i,:);
139.                %随机选取需要进行交叉的点位
140.                d=ceil(rand(narvs/2,1)*(narvs/2));
141.                e=ceil(rand(narvs/2,1)*(narvs/2));
142.                %对点位进行修正
143.                for j=1:narvs/2
144.                    if d(j)==0
145.                        d(j)=1;
146.                    elseif e(j)==0
147.                        e(j)=1;
148.                    end
149.                end
150.                index=[d,e];
151.                text=zeros(narvs/2,1);
152.                %导入每一组相关系数
153.                for j=1:length(index)
154.                    text(j)=relation(d(j),e(j));
155.                end
156.                table=[text,index];
157.                table=sortrows(table,'descend');
158.                %依据覆盖率按照相关系数进行由大到小的排序
159.                cover=rand;
160.                while cover<0.7
161.                    cover=rand;
162.                end
163.                range=round(narvs*cover/2);
164.                table=table(1:range,:);
165.                for k=1:range
166.                    pick=rand;
```

```
167.                    v1=x(table(k,2));
168.                    v2=x(table(k,2));
169.                    x(table(k,2))=pick*v2+(1-pick)*v1;
170.                    x(table(k,3))=pick*v1+(1-pick)*v2;
171.                end
172.                x=check(x,x_up,x_dw);
173.                chrom(i,:)=x;
174.    %对后半段染色体进行相同操作
175.                x=chrom(i,:);
176.                d=ceil(rand(narvs/2,1)*(narvs/2)+narvs/2);
177.                e=ceil(rand(narvs/2,1)*(narvs/2)+narvs/2);
178.                for j=1:narvs/2
179.                    if d(j)==0
180.                        d(j)=1;
181.                    elseif e(j)==0
182.                        e(j)=1;
183.                    end
184.                end
185.                index=[d,e];
186.                text=zeros(narvs/2,1);
187.                d=d-narvs/2;
188.                e=e-narvs/2;
189.                for j=1:narvs/2
190.                    if d(j)==0
191.                        d(j)=1;
192.                    elseif e(j)==0
193.                        e(j)=1;
194.                    end
195.                end
196.                for j=1:length(index)
197.                    text(j)=relation(d(j),e(j));
198.                end
199.                table=[text,index];
200.                table=sortrows(table,'descend');
201.                cover=rand;
202.                while cover<0.7
203.                    cover=rand;
204.                end
205.                range=round(narvs*cover/2);
206.                table=table(1:range,:);
207.                for k=1:range
208.                    pick=rand;
209.                    v1=x(table(k,2));
210.                    v2=x(table(k,2));
```

```
                    x(table(k,2))=pick*v2+(1-pick)*v1;
                    x(table(k,3))=pick*v1+(1-pick)*v2;
                end
                x=check(x,x_up,x_dw);
                chrom(i,:)=x;
            else
                continue;
            end
        new_chrom=chrom;
    end
    end

    ga_cal.m
    clc,clear;
    %% 导入变量
    load ch.mat;
    load dis.mat;
    load name.mat;
    load min_max.mat;
    load rate.mat;

    %% 设置选取变量
    narvs=round(6*rand+26);
    choice=randperm(49);
    choice=choice(1:narvs);
    new_name=name{choice,1};
    label=name{choice,2};
    sale=name{choice,3};
    relation=corrcoef(repmat(sale,1,narvs));

    %% 设定参数
    gen=300;  %迭代次数
    mu=0.005;   %变异率
    cro=0.95;   %交叉率
    num=100;   %初代种群个数
    x_up=[min_max(choice,1)+26;repmat(26.7,narvs,1)]; %变量上下限
    x_dw=[min_max(choice,2)+26;zeros(narvs,1)];
       %变量个数
    narvs=2*narvs;
    a=[];

    %% 初始化
    chrom=zeros(num,narvs);
    fitness_best_all=zeros(gen,1);
```

```
fitness_ave=zeros(gen,1);
for i=1:num
    for j=1:narvs
        chrom(i,j)=x_dw(j)+(x_up(j)-x_dw(j))*rand(1);
    end
end
fitness=fitnesscal(chrom,label,ch,dis,rate,x_dw,x_up,choice,sale);
fitness_best=max(fitness);
fitness_best_all(1)=fitness_best;
fitness_ave(1)=sum(fitness)/length(fitness);

%% 种群开始进行繁衍
for iter=2:gen
    %进行选择
    [n,~]=size(chrom);
    chrom=ga_sel(fitness,chrom,n);
    %进行交叉
    chrom=ga_cro(chrom,cro,n,narvs,x_dw,x_up,relation);
    cro=cro-0.0000001*iter;
    %变异操作
    chrom=ga_mut(chrom,mu,n,narvs,x_up,x_dw);
    %计算适应度
    fitness=fitnesscal(chrom,label,ch,dis,rate,x_dw,x_up,choice,sale);

    fitness_max=max(fitness);
    %选取每轮最佳适应度
    if fitness_max>=fitness_best
        fitness_best=fitness_max;
        a=find(fitness==fitness_best);
        best_chrom=chrom(a,:);
    end
    fitness_best_all(iter)=fitness_max;
    fitness_ave(iter)=sum(fitness)/length(fitness);
end

%% 显示结果
hold on
plot(1:gen,fitness_best_all-4600,'r')
plot(1:gen,fitness_ave-2650,'b')
hold off
legend(['平均适应度';'最佳适应度']);
disp(['最大值为',num2str(1/fitness_best)])
disp('对应解为')
disp(best_chrom)
```

```
298.
299.  fitness_cal.m
300.  function profit=fitnesscal(chrom,label,ch,dis,rate,x_up,x_dw,choice,sa
     le)
301.  m=length(label);
302.  [n,~]=size(chrom);
303.  profit=zeros(n,1);
304.  for i=1:n
305.      income=0;
306.      discount=0;
307.      for j=1:m
308.          if label(j)==1
309.              income=income+obj_fun1(chrom(i,j))*sale(j);
310.              discount=discount+(chrom(i,j+m)-
     chrom(i,j))*rate(choice(j))*dis(choice(j));
311.          elseif label(j)==2
312.              income=income+obj_fun2(chrom(i,j))*sale(j);
313.              discount=discount+(chrom(i,j+m)-
     chrom(i,j))*rate(choice(j))*dis(choice(j));
314.          elseif label(j)==3
315.              income=income+obj_fun3(chrom(i,j))*sale(j);
316.              discount=discount+(chrom(i,j+m)-
     chrom(i,j))*rate(choice(j))*dis(choice(j));
317.          elseif label(j)==4
318.              income=income+obj_fun4(chrom(i,j))*sale(j);
319.              discount=discount+(chrom(i,j+m)-
     chrom(i,j))*rate(choice(j))*dis(choice(j));
320.          elseif label(j)==5
321.              income=income+obj_fun5(chrom(i,j))*sale(j);
322.              discount=discount+(chrom(i,j+m)-
     chrom(i,j))*rate(choice(j))*dis(choice(j));
323.          elseif label(j)==6
324.              income=income+obj_fun6(chrom(i,j))*sale(j);
325.              discount=discount+(chrom(i,j+m)-
     chrom(i,j))*rate(choice(j))*dis(choice(j));
326.          end
327.      end
328.      cheng=chrom(i,m+1:2*m)*ch(choice);
329.      profit(i)=income-cheng+discount;
330.  end
331.
332.  datacal_sumdan.m
333.  %此段代码用于计算所选 30 种单品的历史总销量
334.  clc,clear;
```

```
sale_all=[];
for ii=1:30
    %读取数据
    data=[xlsread('F:\2023年国赛\单类总计.xlsx',ii,'A:A'),xlsread('F:\2023年国赛\单类总计.xlsx',ii,'F:F')];
    sale=zeros(1095,1);
    [m,n]=size(data);
    flag=1;
    %开始计算
    for i=1:m
        if data(i,1)==flag
            sale(flag,1)=data(i,2)+sale(flag,1);
        else
    %利用标志变量进行检测，在防止错误发生的同时也可提高运行速度
            while data(i,1)~=flag
                flag=flag+1;
                if data(i,1)==flag
                    sale(flag,1)=data(i,2)+sale(flag,1);
                    break;
                end
            end
        end
    end
    %结果得到总销量
    sale_all=[sale,sale_all];
end

%% 异常值处理
[p,q]=size(sale_all);
miu=zeros(1,30);
sig=zeros(1,30);
high=0;
low=0;
%利用3σ原则处理异常值
for i=1:q
    for j=1:p
    %计算数据平均数和方差
    miu(i)=mean(sale_all(:,i));
    sig(i)=std(sale_all(:,i),0);
    if sale_all(j,i)>miu(i)+3*sig(i)
        high=high+1;
        sale_all(j,i)=randi([0 100]);
    elseif sale_all(j,i)<miu(i)-3*sig(i)
        low=low+1;
```

```
            sale_all(j,i)=rand(1)*5;
        end
        end
    end

    %% 绘制六种单品的销量变化曲线
    index=[1 6 9 15 17 30];
    %设置颜色 RGB 参数
    color=[0.929 0.694 0.125;
        0.494 0.184 0.556;
        0.85 0.325 0.098;
        0.466 0.674 0.188;
        0 0.447 0.741;
        0.301 0.745 0.933];
    text={'金针菇','紫茄子','芜湖青椒','西兰花','静藕','云南生菜'};
    %绘制图像
    for i=1:6
        subplot(2,3,i);
        plot(1:1095,sale_all(:,i),'color',color(i,:));
        legend(text{i})
    end

    datacal_sum.m
    clc,clear;
    %% 计算各个单类总销量
    sale_all=[];
    for ii=1:6
        %加载数据
        data=[xlsread('F:\2023 年国赛\附件
    2.xlsx',ii+1,'A:A'),xlsread('F:\2023 年国赛\附件 2.xlsx',ii+1,'F:F')];
        %初始化
        sale=zeros(1095,1);
        [m,n]=size(data);
        flag=1;
        %开始计算各个大类的历史销售总额
        for i=1:m
            if data(i,1)==flag&&data(i,2)>=0
                sale(flag,1)=data(i,2)+sale(flag,1);
            elseif data(i,1)==flag&&data(i,2)<0
                sale(flag,1)=data(i,2)+sale(flag,1);
            elseif data(i,1)~=flag&&data(i,2)>=0
                flag=flag+1;
                sale(flag,1)=data(i,2)+sale(flag,1);
            elseif data(i,1)~=flag&&data(i,2)<0
```

```
                flag=flag+1;
                sale(flag,1)=data(i,2)+sale(flag,1);
            end
        end
        sale_all=[sale,sale_all];
    end

    %% 去除异常值
    [p,q]=size(sale_all);
    miu=zeros(1,6);
    sig=zeros(1,6);
    high=0;
    low=0;
    %利用 3σ 原则进行异常值处理
    for i=1:q
        for j=1:p
        miu(i)=mean(sale_all(:,i));
        sig(i)=std(sale_all(:,i),0);
        if sale_all(j,i)>miu(i)+3*sig(i)
            high=high+1;
            sale_all(j,i)=miu(i)+3*sig(i);
        elseif sale_all(j,i)<miu(i)-3*sig(i)
            low=low+1;
            sale_all(j,i)=miu(i)-3*sig(i);
        end
        end
    end

    %% 绘制有季度分割线的图像
    text={'根茎类销量';'食用菌类销量';'辣椒类销量';'花叶类销量';'花菜类销量';'茄子类销量'};
    ymax=[300 600 700 1400 200 120]*0.7;
    point=sale_all(91:91:1092,:);
    for k=1:6
        hold on
        figure(k)
        plot(1:1095,sale_all(:,k),'b')
        line([365 365],[0 ymax(k)],'color','g','linewidth',2)
        line([730 730],[0 ymax(k)],'color','c','linewidth',2)
        line([91 91],[0 ymax(k)*0.8],'linestyle','--','linewidth',1.5,'color','r')
        line([182 182],[0 ymax(k)*0.8],'linestyle','--','linewidth',1.5,'color','r')
```

```
461.        line([273 273],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
462.        line([455 455],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
463.        line([546 546],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
464.        line([637 637],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
465.        line([819 819],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
466.        line([910 910],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
467.        line([1001 1001],[0 ymax(k)*0.8],'linestyle','--
    ','linewidth',1.5,'color','r')
468.        hold off
469.        %设置图像参数和标签
470.        axis([0 1100 0 ymax(k)])
471.        xlabel('日期代码（单位：天 ）')
472.        ylabel('日总销量（单位：kg）')
473.        legend(text{k},'第一周年分界线','第二周年分界线','季度分界点')
474.    end
475.
476.    %% 绘制无季度分割线的图像
477.    text={'根茎类销量';'食用菌类销量';'辣椒类销量';'花叶类销量';'花菜类销量';'
    茄子类销量'};
478.    ymax=[300 600 700 1400 200 120]*0.7;
479.    point=sale_all(91:91:1092,:);
480.    for k=1:6
481.        hold on
482.        figure(k)
483.        plot(1:1095,sale_all(:,k),'b')
484.        line([365 365],[0 ymax(k)],'color','g','linewidth',2)
485.        line([730 730],[0 ymax(k)],'color','c','linewidth',2)
486.        hold off
487.        %设置图像参数和标签
488.        axis([0 1100 0 ymax(k)])
489.        xlabel('日期代码（单位：天 ）')
490.        ylabel('日总销量（单位：kg）')
491.        legend(text{k},'第一周年分界线','第二周年分界线')
492.    end
493.
494.    data_income.m
495.    %此处代码用于计算总收入
496.    clc,clear;
```

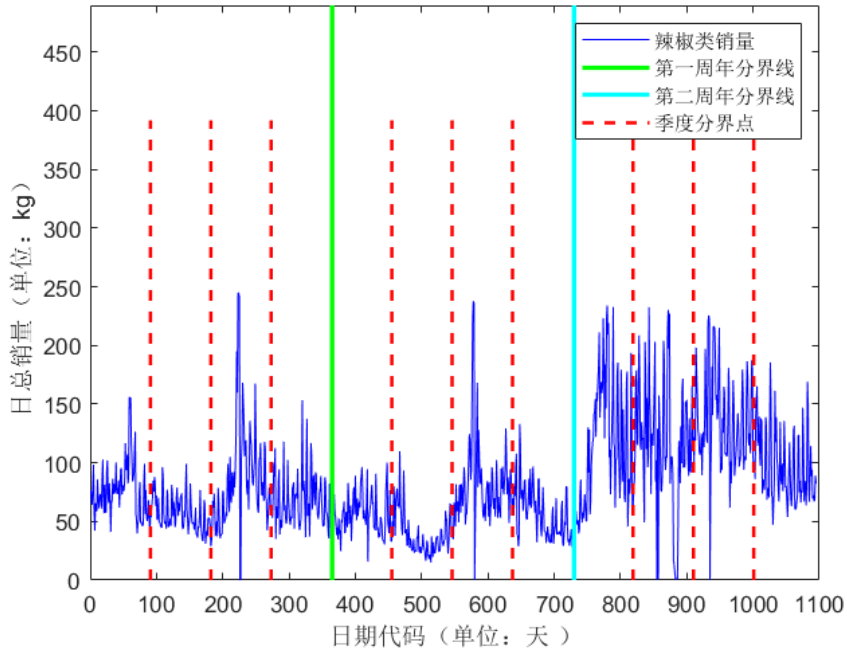
```
income_all=[];
for ii=1:6
    data=[xlsread('F:\2023 年国赛\附件 2.xlsx',ii+1,'A:A'),xlsread('F:\2023 年国赛\附件 2.xlsx',ii+1,'F:F'),xlsread('F:\2023 年国赛\附件 2.xlsx',ii+1,'G:G')];
    income=zeros(1095,1);
    [m,n]=size(data);
    flag=1;
    for i=1:m
        if data(i,1)==flag
            income(flag,1)=data(i,2)*data(i,3)+income(flag,1);
        else
            while data(i,1)~=flag
                flag=flag+1;
                if data(i,1)==flag
                    income(flag,1)=data(i,2)*data(i,3)+income(flag,1);

                    break;
                end
            end
        end
    end
    income_all=[income_all,income];
end

%%
load sale_all_sig.mat
load income_all.mat
income_all=income_all(366:1095,:);
sale_all=sale_all(366:1095,:);
price=income_all./sale_all;
[p,q]=size(income_all);
miu=zeros(1,6);
sig=zeros(1,6);
high=0;
low=0;
for i=1:q
    for j=1:p
    miu(i)=mean(income_all(:,i));
    sig(i)=std(income_all(:,i),0);
    if income_all(j,i)>miu(i)+3*sig(i)
        high=high+1;
        income_all(j,i)=miu(i)+3*sig(i);
    elseif income_all(j,i)<miu(i)-3*sig(i)
```

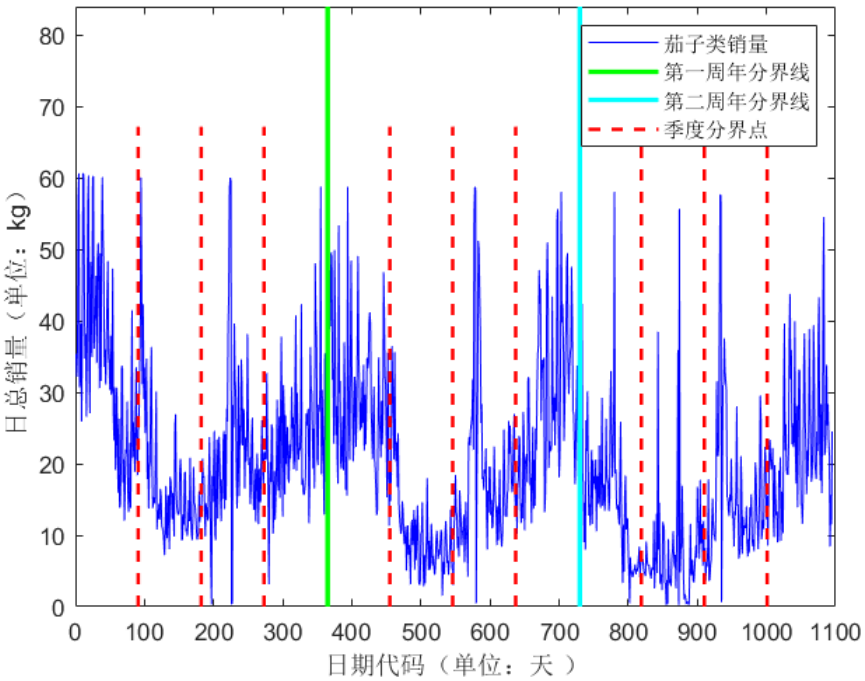
```
            low=low+1;
            income_all(j,i)=miu(i)-3*sig(i);
        end
        end
    end
    pick=0.15;
    index=[];
    for i=1:length(price)
        if rand<pick
            index=[index,i];
        end
    end
    price=price(index,:);
    sale_all=sale_all(index,:);
    x1=sale_all(:,1);
    x2=sale_all(:,2);
    x3=sale_all(:,3);
    x4=sale_all(:,4);
    x5=sale_all(:,5);
    x6=sale_all(:,6);
    y1=price(:,1);
    y2=price(:,2);
    y3=price(:,3);
    y4=price(:,4);
    y5=price(:,5);
    y6=price(:,6);
    %% 绘制拟合散点图
    v=5:0.1:125;
    w=(-4.009*v.^2+1447*v-1735)./(v.^2+149.7*v-554.3);
    hold on
    plot(v,w);
    scatter(x1,y1,'r*')
    hold off

    %% 绘制拟合散点图
    v=3:0.1:125
    y=(-4.988*v.^2+476.7*v-305.4)./(v.^2+1.389*v-1.679);
    hold on
    plot(v,y);
    scatter(x6,y6,'r*');
    axis([-10 105 -10 70])

    datacal_chb.m
    %此处代码用于计算成本
```

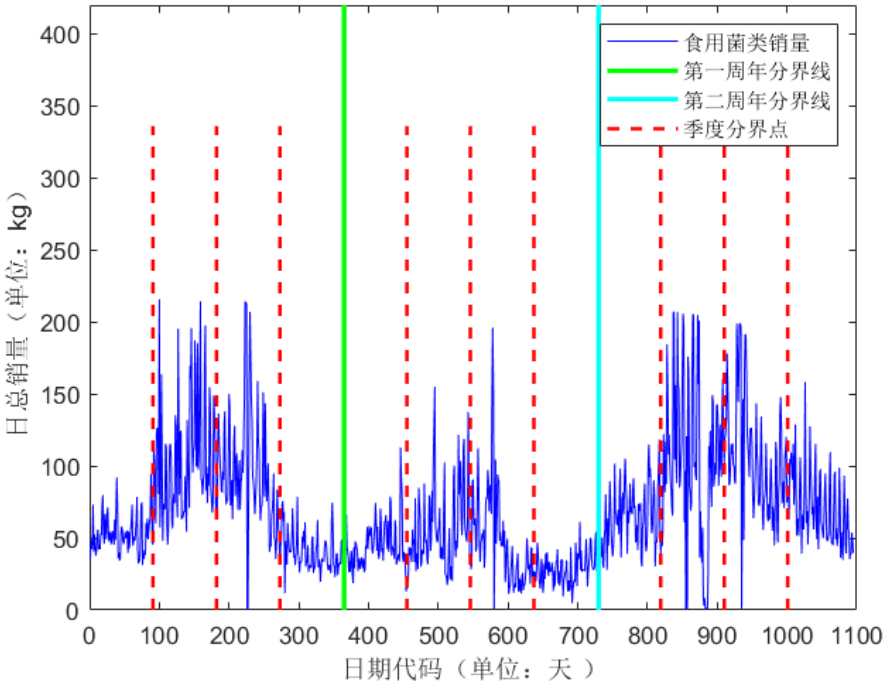
```
clc,clear;
price_all=[];
mass_all=[];
for ii=1:6
    data=xlsread('F:\2023 年国赛\附件 3.xlsx',ii+2);
    price=zeros(30,1);
    mass=zeros(30,1);
    [m,n]=size(data);
    flag=1;
    for i=1:m
        if data(i,1)==flag
            price(flag,1)=data(i,2)+price(flag,1);
            mass(flag,1)=data(i,3)+mass(flag,1);
        else
            while data(i,1)~=flag
                flag=flag+1;
                if data(i,1)==flag
                    price(flag,1)=data(i,2)+price(flag,1);
                    mass(flag,1)=data(i,3)+mass(flag,1);
                    break;
                end
            end
        end
    end
    price_all=[price_all,price];
    mass_all=[mass_all,mass];
end
chb_all=price_all./mass_all;

check.m
%此函数用于检查染色体范围是否越界
function result=check(x,x_up,x_dw)
[m,n]=size(x);
for i=1:m
    for j=1:n
        if x(i,j)>x_up(j)              %检查每个数是否越界
            x(i,j)=x_up(j);
        elseif x(i,j)<x_dw(j)
            x(i,j)=x_dw(j);
        end
    end
result=x;
 end
```

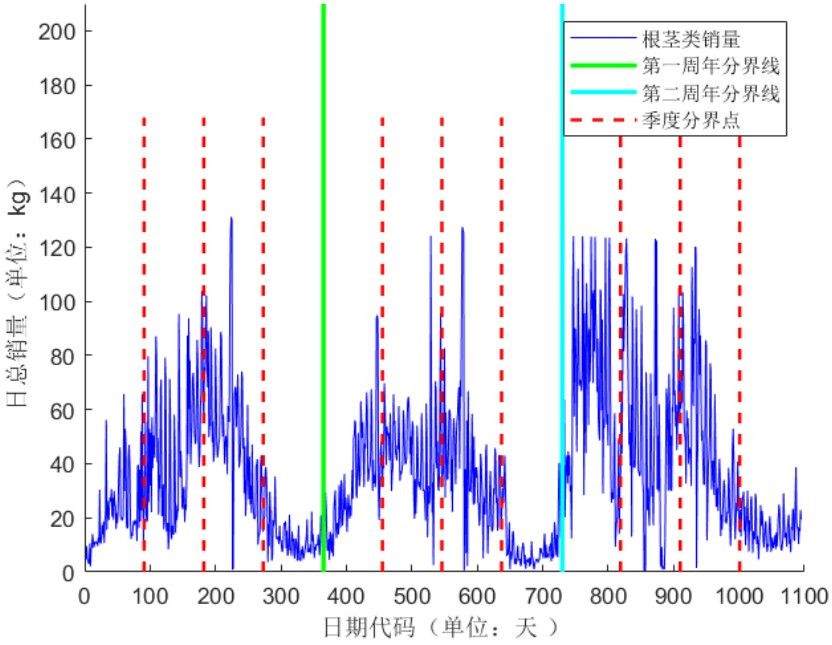
附录 2

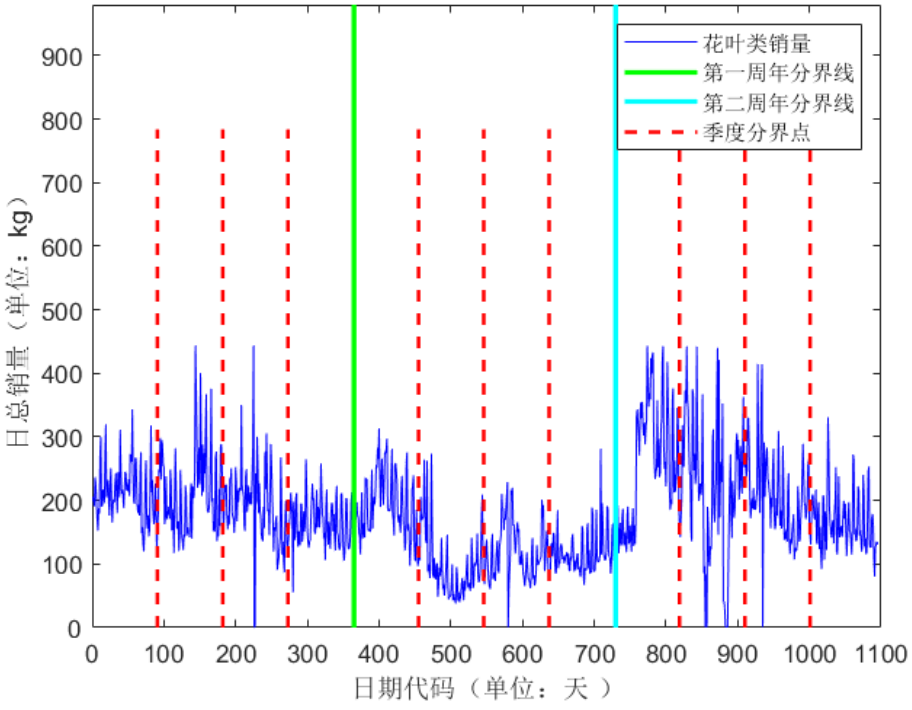
相关图片

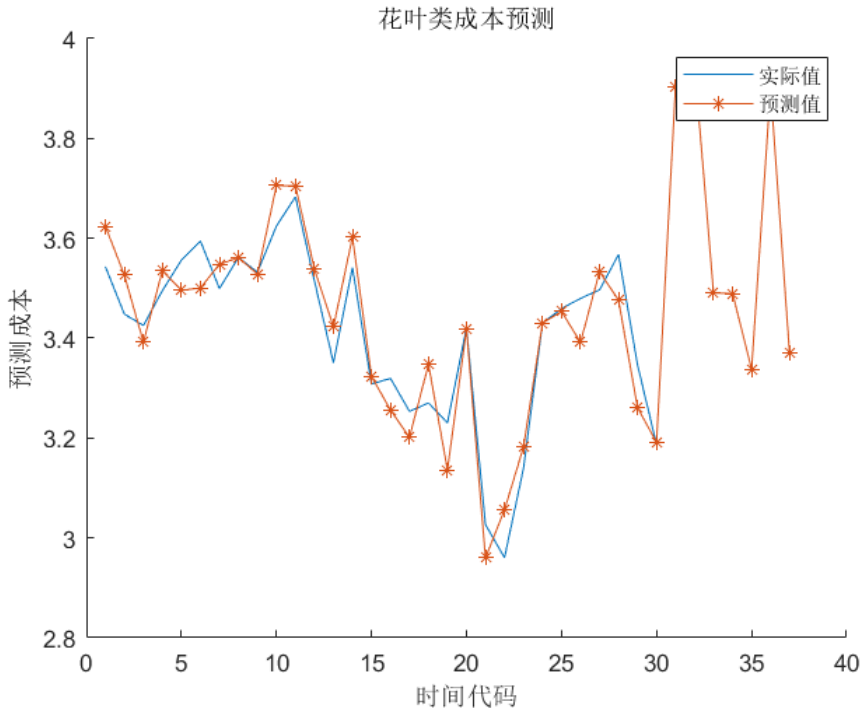

辣椒类销量
第一周年分界线
第二周年分界线
季度分界点
日总销量（单位：kg）
日期代码（单位：天）
0
50
100
150
200
250
300
350
400
450
0
100
200
300
400
500
600
700
800
900
1000
1100

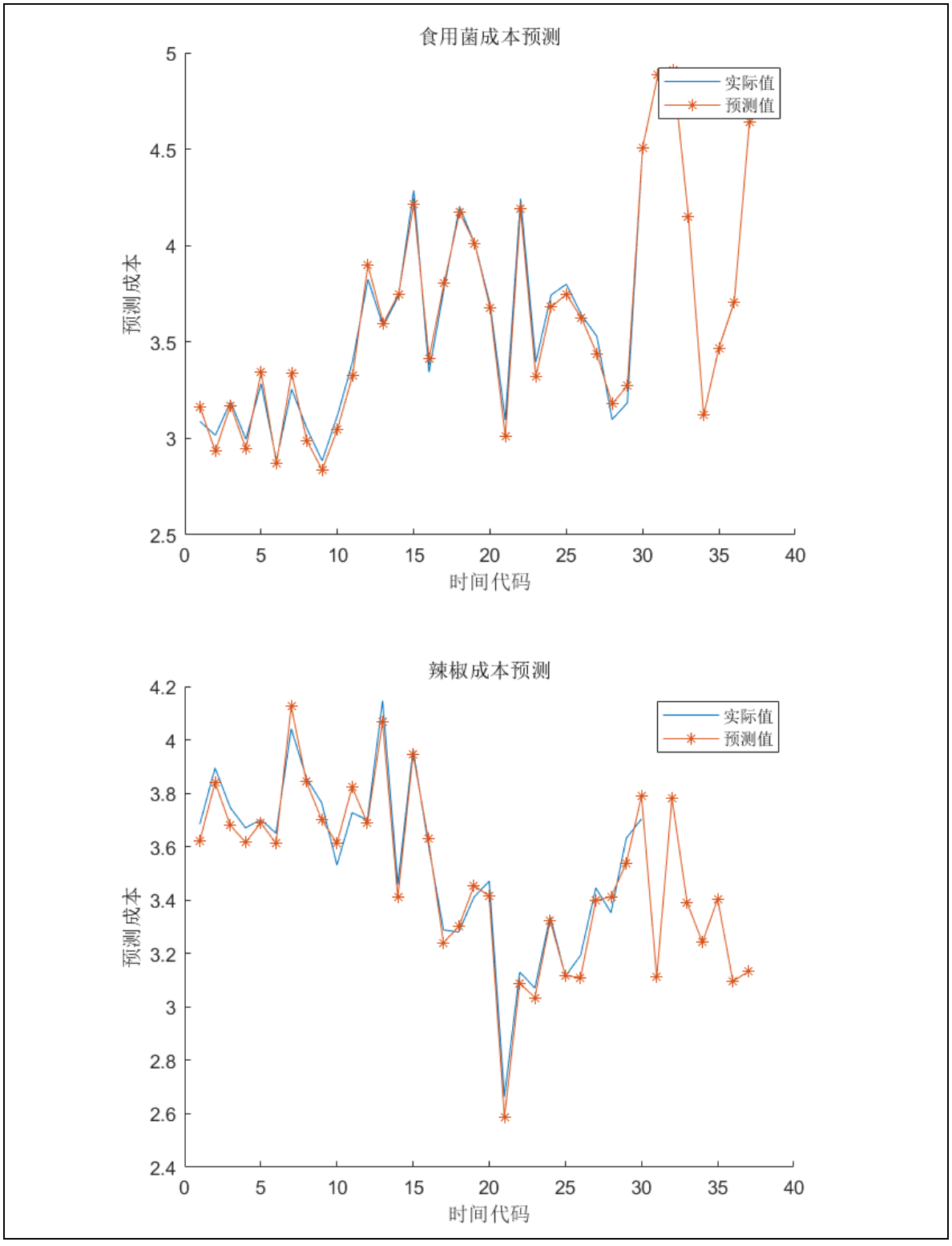

茄子类销量
第一周年分界线
第二周年分界线
季度分界点
日总销量（单位：kg）
日期代码（单位：天）
0
10
20
30
40
50
60
70
80
0
100
200
300
400
500
600
700
800
900
1000
1100

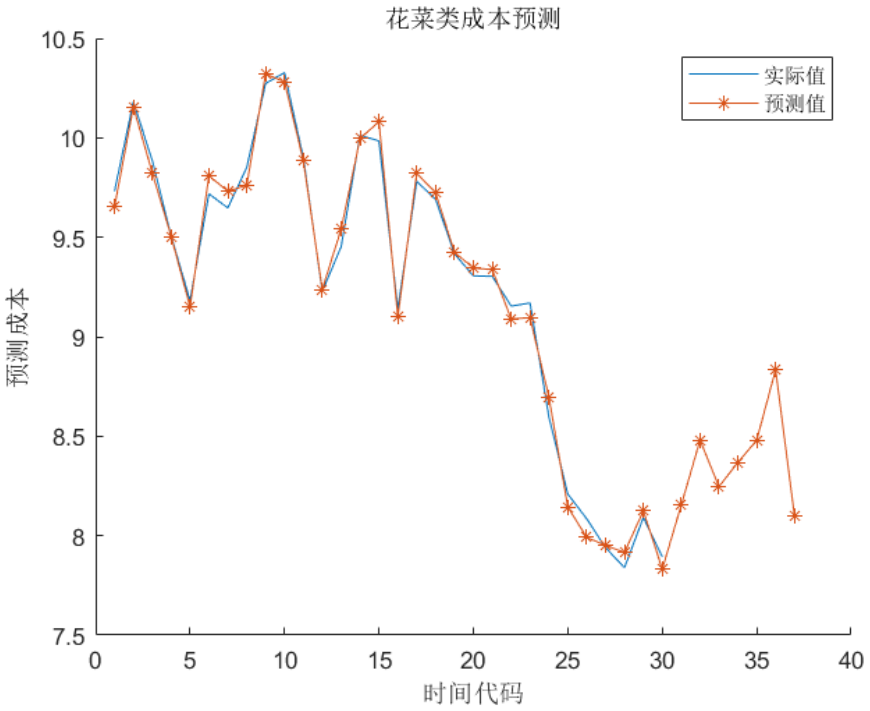

食用菌类销量
第一周年分界线
第二周年分界线
季度分界点
日总销量（单位：kg）
日期代码（单位：天）
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700
800
900
1000
1100

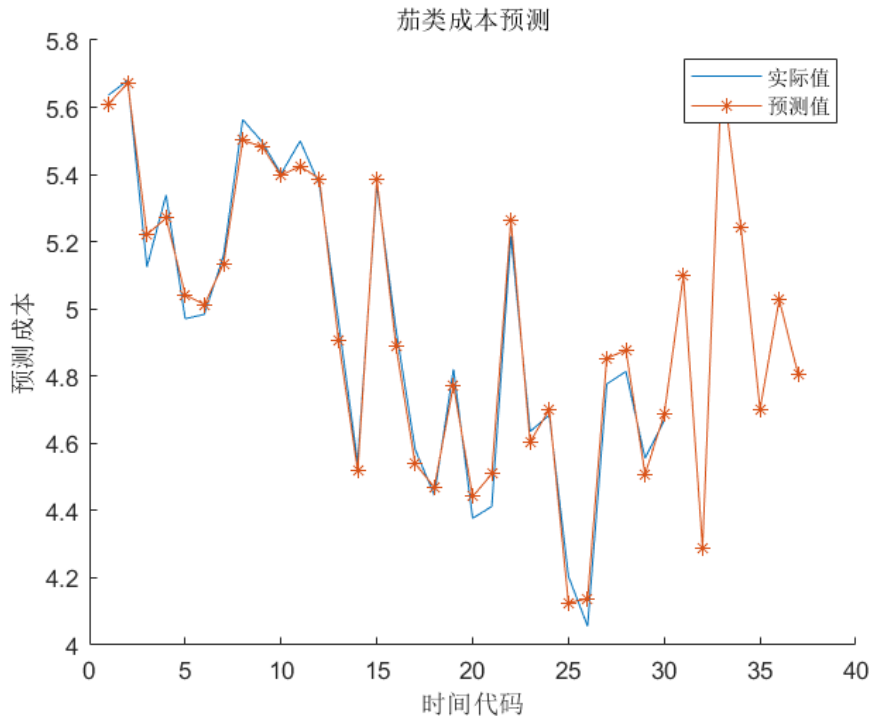

根茎类销量
第一周年分界线
第二周年分界线
季度分界点
日总销量（单位：kg）
日期代码（单位：天）
0
20
40
60
80
100
120
140
160
180
200
0
100
200
300
400
500
600
700
800
900
1000
1100


花叶类销量
第一周年分界线
第二周年分界线
季度分界点
日总销量（单位：kg）
日期代码（单位：天）
0
100
200
300
400
500
600
700
800
900
1000
1100


花叶类成本预测
实际值
预测值
预测成本
时间代码
2.8
3
3.2
3.4
3.6
3.8
4
0
5
10
15
20
25
30
35
40


食用菌成本预测
实际值
预测值
预测成本
时间代码
辣椒成本预测
实际值
预测值
预测成本
时间代码





附录 3
介绍：该代码是某某语言编写的，作用是什么