

基于预测回归规划模型的蔬菜自动定价与补货决策

摘 要

蔬菜类的保鲜期一般较短，且品相会随时间流逝而下降，因此，做出正确的蔬菜每日补货量与定价决策，是商超盈利的重中之重。本文从蔬菜的销售量和批发价出发，基于数学模型探究蔬菜定价与补货的最佳决策方法。

针对问题一，本文对六大单品及其各单品的销售情况进行描述统计，并通过制作饼图、折线图等将数据**可视化**，分析其分布规律，得出花叶类、辣椒类销售占比较大且销售量较不稳定。通过对各品类销售量进行**正态分布检验**，对其进行**皮尔逊相关系数分析**，可得花菜类和花叶类的相关性最为显著，茄类与其他品类均相关性不显著。各单品之间的规律和关系则通过销售频率图展示，最终得到辣椒类的芜湖青椒销售频率最高，花叶类的芥兰和辣椒类的水果辣椒(橙色)销售频率最低。

针对问题二，本文计算得到各品类的日均批发价、损耗量，并进行 **Arima 时间序列分析**预测未来一周的各品类日批发价。随后利用日均售价与日均销售额建立**非线性回归模型**并进行**多项式拟合**，通过拟合优度调试参数得到拟合度最高的两者关系。最后以批发价、损耗量、日销售量为限制条件对各品类进货量与定价建立**非线性规划模型**，得到商超未来一周各品类的具体最优进货量、定价（详细结果见 5.2.5 非线性规划求最优决策），商超最大收益为 10416.2648 元。

针对问题三，首先对 6 月 24 日-30 日的单品售卖情况进行描述统计，获得各单品逐日销量、利润、批发价，再将其作为自变量建立**线性回归预测模型**，预估出 7 月 1 日的单品日均批发价和销售量数据。随后将三项数据与单品总数约束条件、陈列量约束条件作为参数构成目标函数，建立**粒子群算法模型**，通过迭代，测算目标函数最优解，最终形成各单品最优补货决策（详细结果见附录五），商超最大收益为 750 元。

针对问题四，本文从地域、目标人群、市场规律、时事新闻等方面探究外因对消费意愿、蔬菜批发价的影响，在问题一信息处理的基础上，额外采集周围超商位置客源及蔬菜商品种类供应量、该超商的客户群体种类数量信息数据、该商超地蔬菜库存实时监测数据、批发地的距离品种数量品质数据、相关蔬菜商品的网络时事热点趋势动向数据五类数据信息的决定。

最后，本文对模型进行了优缺点评价和灵敏度分析，对蔬菜定价与补货量有了更深刻的探究与理解。

关键词：时间序列分析 回归预测 多项式拟合 非线性规划 补货量 粒子群算法

一、问题重述

一般蔬菜类商品的新鲜期较短，且随出售时间的增长而品质下降。因此，商超需要结合各蔬菜具体的供需关系进行补货。

由于蔬菜的种类众多、产地多样化，且进货时间基本在凌晨 3:00-4:00，商超需要在没有明确具体单品和进价的情形中做出补货决策。“成本加成定价”法是产品单位成本加上一定比例的利润制定产品价格的方法，是常用的蔬菜定价方法。为根据供需关系形成合力的销售组合，对市场需求的有效分析以及补货和定价的正确决策显得尤为重要。

在本题中，题目给出了 6 个蔬菜品类批发价损耗率等基本信息以及其 2020 年 7 月 1 日至 2023 年 6 月 30 日的销售信息，要求根据实际情况建立数学模型，解决以下问题：

1. 不同品类或不同单品的蔬菜类商品之间可能存在关系，探究各品类以及单品销售量的分布规律和关系。
2. 探究各品类蔬菜的销售总量与售价的关系，并以品类为单位计划未来一周（2023 年 7 月 1-7 日）的日补货总量和定价决策，使利润最大。
3. 由于蔬菜类商品市场空间的限制，要求根据 2023 年 6 月 24-30 日的蔬菜量，制定 7 月 1 日蔬菜单品补货和定价决策，在满足单品总数置于 27-33 个之间，且各单品订购量 大于 2.5 千克的条件下，使利润最大。
4. 为了优化蔬菜商品的补货和定价决策，补充商超需要采集的其他数据类型，并给出理由。

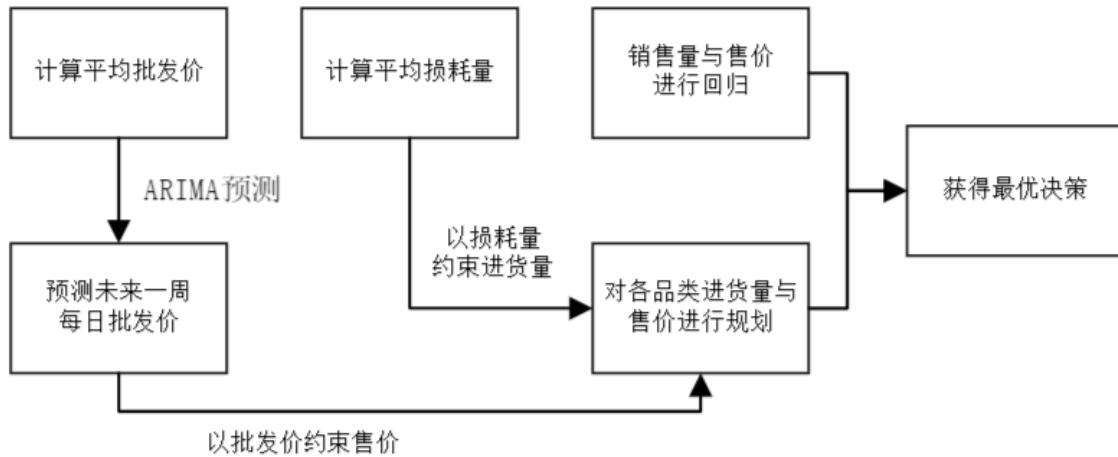
二、问题分析

2.1 问题一的分析

问题一要求探究各品类和单品销售量的分布规律以及相互关系。整合附件一、附件二，获得六大品类以及各单品的销售情况，从而对其进行分析。品类之间销售量的分布关系可从整体占比与不同时间销售量来分析，本文通过制作饼图对各品类整体占比进行可视化分析，并将时间分别按照季度和每日具体时间进行分类，对各品类进行单独分析与对比分析探究其分布规律。通过对品类进行正态分布检验，用皮尔逊相关系数进行相关性分析，获得各品类之间的探究关系，并附热力图直观展示分析。各单品的分布规律及相互关系则通过各单品的销售频率以表示。

2.2 问题二的分析

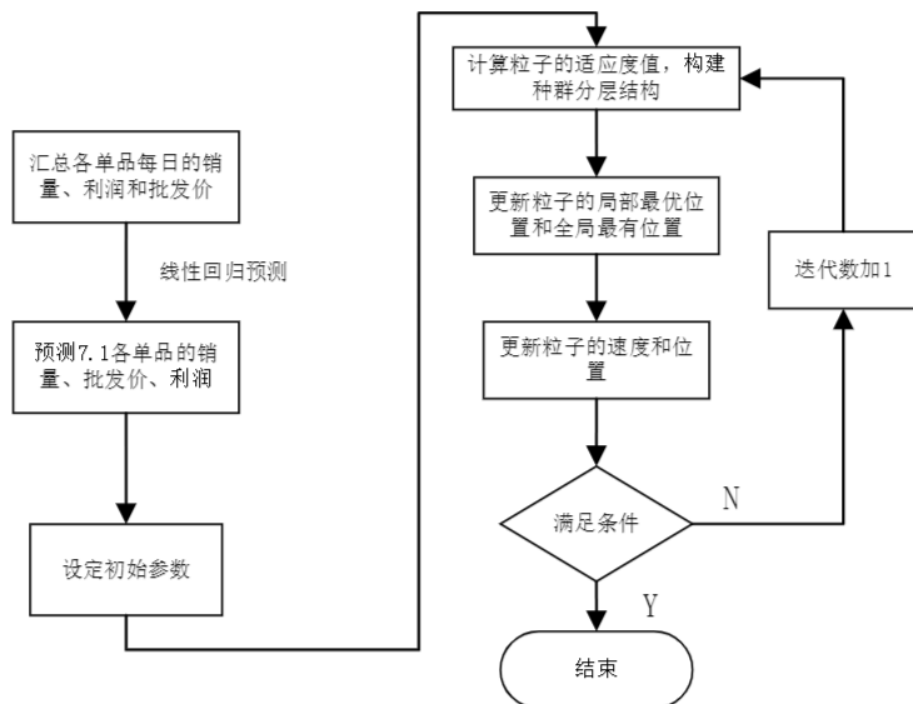
问题二要求以蔬菜品类为单位分析销售总量与售价的关系，并制定未来一周的蔬菜最优日补货计划与定价策略。销售总量与售价相互影响，且销售总量与进货量相关，售价与进价相关。本文首先计算各品类的日平均批发价，通过 ARIMA 模型对其进行预测，获得未来一周的日平均批发价；其次，以日为单位将当日售价、前日销售额与今日销售额回归获得今日销售额与二者的关系；最后计算品类平均损耗率，以批发价和平均损耗率进行约束，对未来一周各品类的进货量与售价进行规划



图一：问题二思路流程图

2.3 问题三的分析

问题三要求在新的约束条件下从单品补货总量和定价策略的维度上进一步制定补货计划，使商超收益最大。根据6月24日-30日的单品售卖情况，汇总出每类单品在该段时间每日的销量、利润、批发价，并作为自变量建立线性回归模型，预估出7月1日的销量、利润以及批发价。该三项数据将和单品补货陈列和单品类数的约束条件作为参数转化为一个目标函数，并在粒子群算法模型中通过粒子不断迭代筛选最后得到最优的目标函数解即最高的利润值。



图二：问题三流程图

2.4 问题四的分析

问题四要求寻找采集的数据以优化蔬菜补货和定价的决策。本文主要集中于补货前的信息采集。在补货前，可根据商超所处的位置人群特征进行信息采集，分析其普

遍需求，并根据库存对不同批发场地进行合理进货。在补货后，可通过对其周围商超信息进行采集，提前做出决策提高市场竞争力。此外，还可采集时事热点，遵循市场规律，及时调整进货方案。

三、模型假设

1. 损耗率中已包含蔬菜因时长、温度、水分等环境不理想而腐败变质的情况。
2. 仅考虑各品类蔬菜之间的相互影响，需排除肉类、禽类等其他食物对蔬菜的影响。
3. 销售量仅占新鲜批发量的 80%。
4. 单品退货率仅与其自身有关，不受时间与其他蔬菜等影响。

四、符号说明

符号	说明
σ	协方差
ρ	皮尔逊相关系数
d_i	求斯皮尔曼相关系数时 X 与 Y 的等级差
r_s	斯皮尔曼相关系数
m	品类日总批发价
n	品类日总进货量
S	利润
a	日销售量
n	品类批发量

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 数据预处理

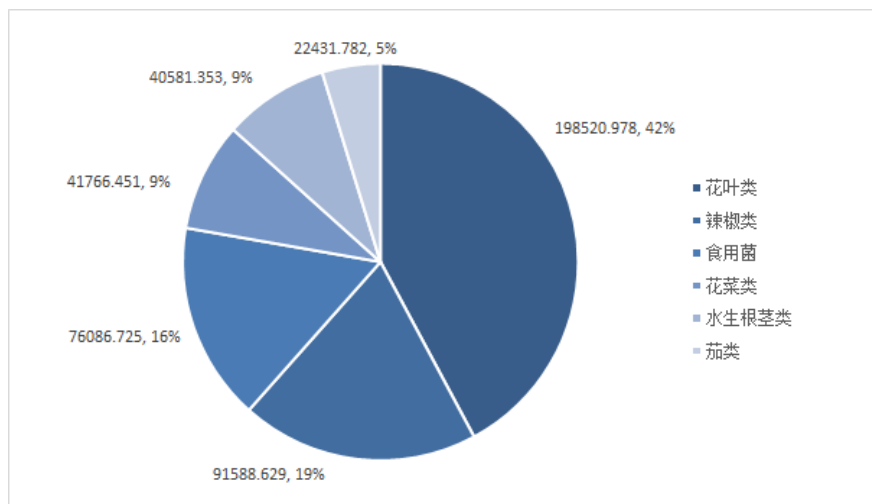
通过“VLOOKUP”函数将附件一与附件二的内容整合，获得各品类和各单品蔬菜的具体销售情况。通过观察数据，本文发现茄类部分日期没有销售额，本文认为有以下两种解释：

1. 当日茄类未进行销售
2. 当日茄类销售数据缺失

其中部分非连续日期的茄类销售量为 0，且前后日期均正常销售，本文将该日期视为当日未进行茄类进货与销售。2022 年 9 月 12 日至 2022 年 10 月 14 日均无茄类销售量，考虑到一个月不进货茄类蔬菜的可能性较小，本文将其视为缺失值，并通过时间序列分析对其进行填补，预测结果图可见附录 6。

5.1.2 各品类销售量分布规律探究

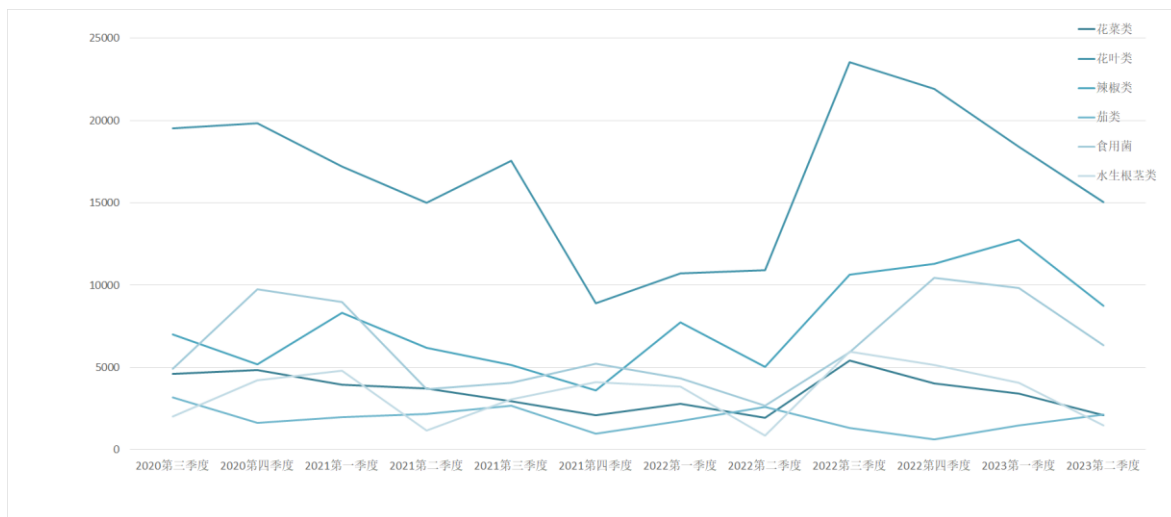
为探究各品类之间销售量的整体关系，本文计算了 2020 年 7 月 1 日至 2023 年 6 月 30 日的各品类总销售量，并获得了如下饼图：



图三：各品类总销售量

由图分析可得：花叶类蔬菜种类丰富，拥有多达 105 种单品，作为一种富含维生素、矿物质和膳食纤维的食物，是一日三餐中的重要食物，占蔬菜消费的主要部分，在过去三年中，拥有最大的净销售量，大约占了蔬菜类销售的一半，说明人们对花叶类蔬菜需求较大；其次是拥有 45 种单品的辣椒类，作为重要的调味品原料之一，辣椒类蔬菜是能起到刺激唾液、胃液分泌从而增强食欲、促进消化的高效功能蔬菜，在销售中也占有较大市场，占比约 22%；所属水生根茎类食物的单品较少，仅拥有 19 种单品，多为藕类，种类较为单一，人们对其的消费意愿相对较低，仅在蔬菜类销售量中占比 4%。

为获得各品类蔬菜销售量与宏观时间的关系，我们先将时间按照四个季度进行分类，将其可视化可得下图：

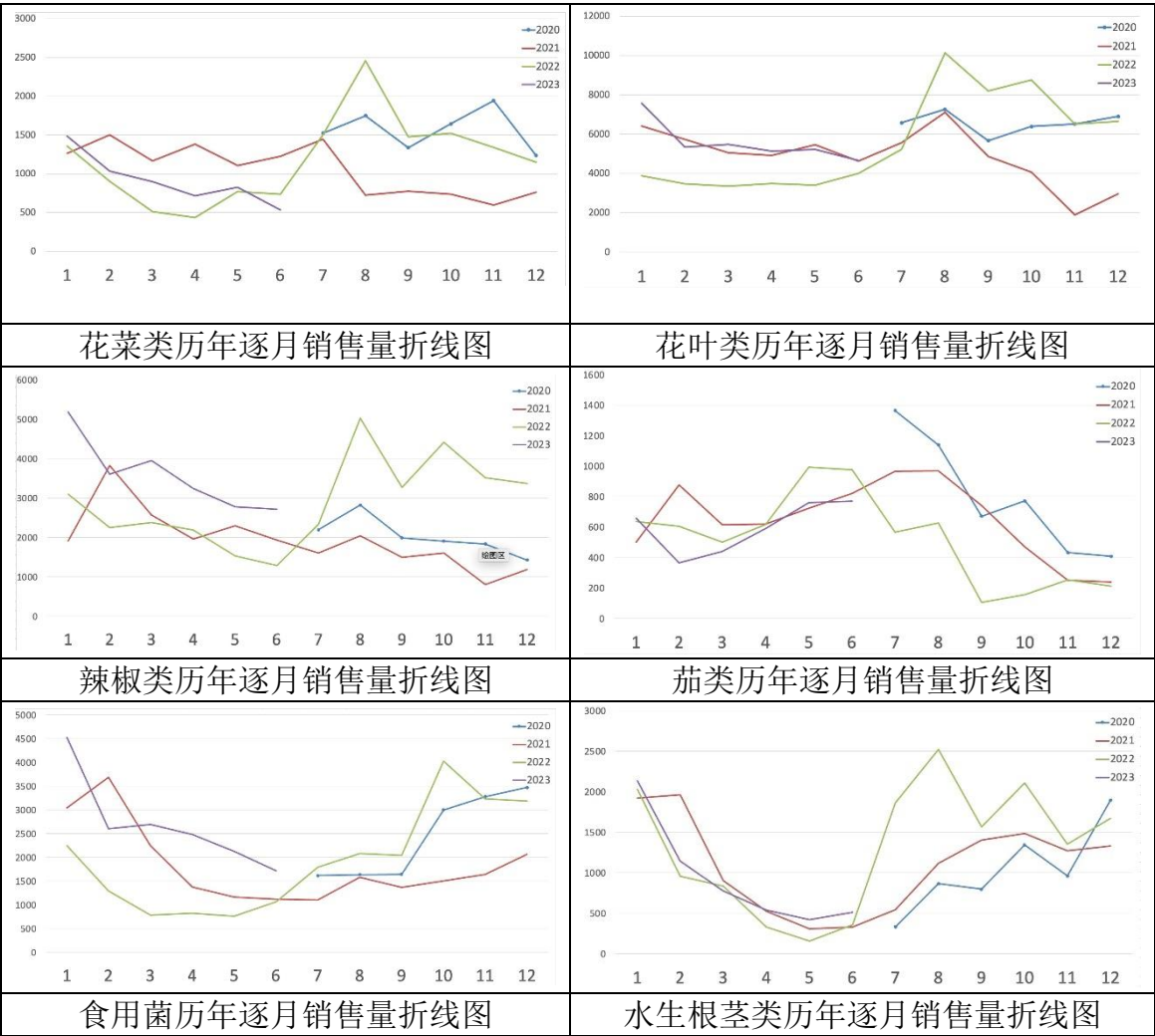


图四：各品类与季度的关系

由图分析可得：辣椒类、茄类、食用菌、水生根茎类均大概呈现以年为单位的波动周期性。其中，2021 年末与 2022 年初的销售量普遍较少。就各品类而言，食用菌、水生根茎类蔬菜波动较为相似，且在第二季度迎来谷值，与其成熟上市时间紧密相连；茄类波动相比之下较为平缓，且一般在第二季度、第三季度销售量较高，可与茄类蔬菜于夏秋成熟相关；辣椒类均在第一季度的销售量较高，可能与冬季季

节寒冷，吃辣椒类食物暖胃驱寒、促进血液循环相关。同时，本文观察到 2022 年末开始各品类蔬菜均呈现不同程度的上升趋势，人们对蔬菜的需求普遍提高，可能与疫情政策开放相关。

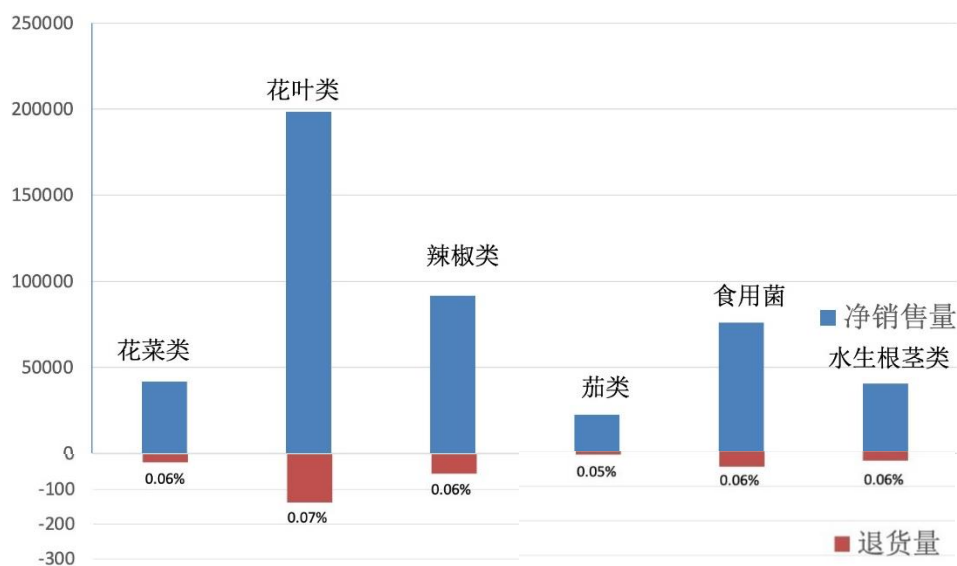
此外，本文将历年逐月的销售量进行可视化，将各品类蔬菜进行纵向对比。



图五：各品类历年逐月销售量

由图分析可得：花菜类、花叶类、辣椒类、茄类、食用菌每月的销售量与往年同月的销售量差异较为明显，月销售量不稳定，受外界影响较大；水生根茎类往年同月销售量基本相同，受外界环境影响较小，人们每年同月对其的需求波动轻微，月销售量较为稳定。

为探究各品类蔬菜销售量与退货量的情况，本文计算获得如下图：

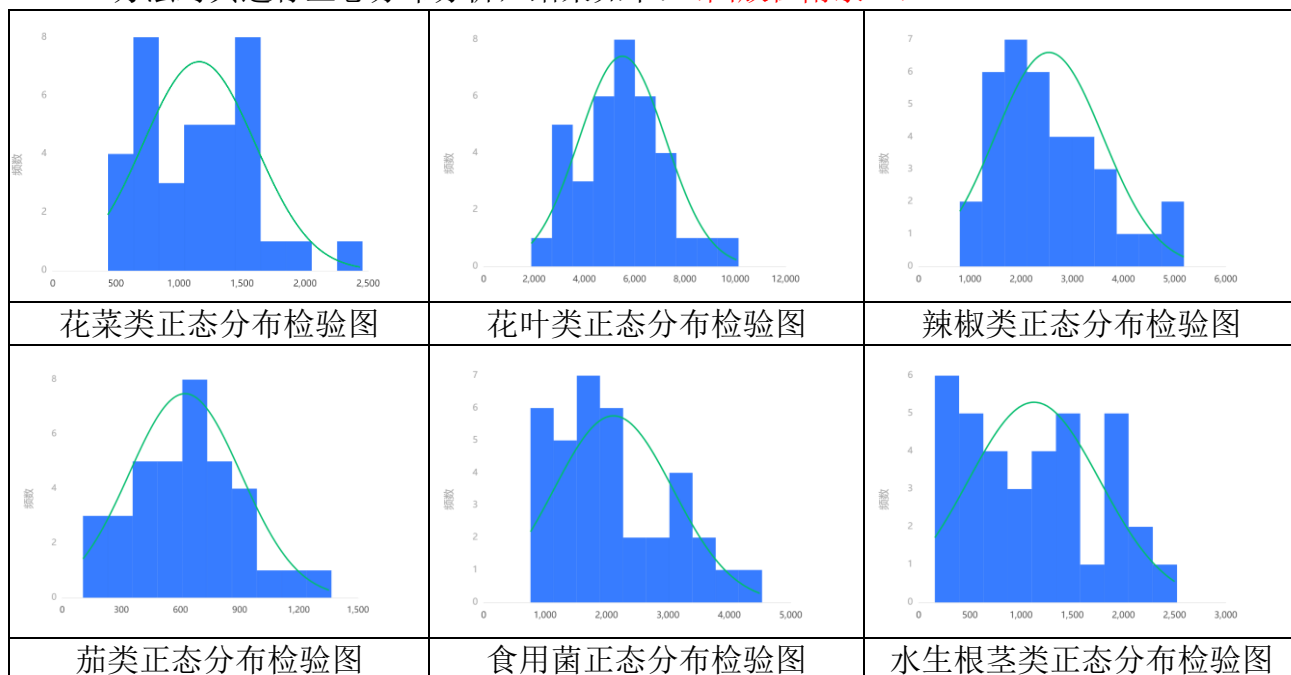


图六：各品类蔬菜销售量与退货量

该图上半部分为各品类的净销售量，下半部分为各品类的退货量，数字为其占比。由图可得，不同品类之间的退货率基本相同，大约都在 0.06% 左右。

5.1.3 各品类销售量相互关系模型建立

为探究各品类之间的相互联系，本文先对其进行相关性分析。首先用 Kolmogorov-Smirnov 方法对其进行正态分布分析，结果如下：（图放在附录？）



图七：各品类正态分布检验图

由图可得，六种品类基本上呈现出钟形，可接受为正态分布，故对此进行皮尔逊相关系数分析。皮尔逊系数用于测量两个变量之间的线性相关性，其值在 -1 和 1 之间，其中 1 表示完全正相关，-1 表示完全负相关，0 表示无线性相关。

本文以花菜类和花叶类相关系数为例。令花菜类三年来各个月份销售量为集合 $X: \{x_1, x_2, \dots, x_{36}\}$ ，花叶类三年来各个月份销售量为集合 $Y: \{y_1, y_2, \dots, y_{36}\}$ ，首先根据以下公式算得 X 、 Y 均值和协方差：

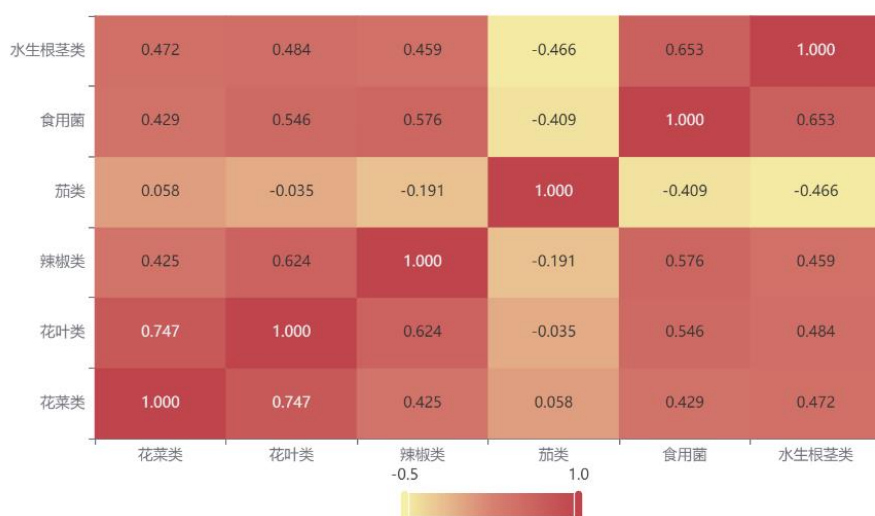
$$E(X) = \frac{\sum_{i=1}^{36} X_i}{36} \quad (1)$$

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{36} (X_i - E(X))^2}{36}} \quad (2)$$

从而可根据以下公式获得花菜类和花叶类皮尔逊相关系数：

$$\rho_{XY} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{36} \frac{(X_i - E(X))}{\sigma_X} \frac{(Y_i - E(Y))}{\sigma_Y}}{36} \quad (3)$$

以此类推，可获得六大品类两两之间的皮尔逊相关系数，最终获得如下热力图



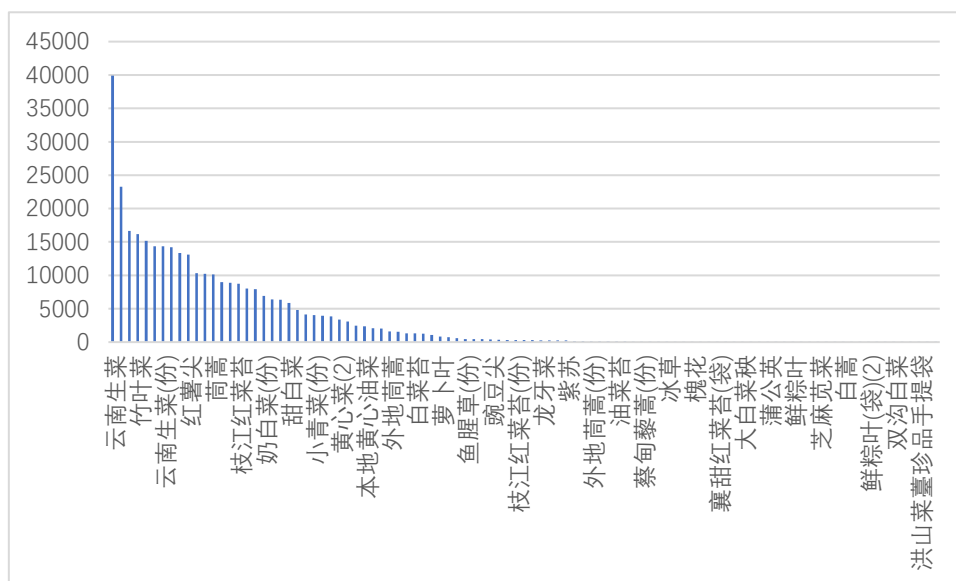
图八：各品类皮尔逊系数热力图

由图可得：花菜类、花叶类与其他各品类之间相关性较大，其中花菜类和花叶类的相关性最为显著，且二者的变化趋势较为接近；茄类与其他各品类之间的相关性均不太显著，且基本与其他品类呈现的变化趋势不同。

5.1.4 不同单品销售量分布规律与相互关系探究

本文通过探究各单品历年消费频率分析其销售量的分布规律，此处以花叶类单品为例。

通过统计分析，可获得花叶类各单品的频率如下图（其他品类各单品频率见附录1）：



图九：花叶类各单品消费频率

由图可得，云南生菜的消费频率较高，消费需求较大，购买人次较多，其次是云南油麦菜，由此可分析云南蔬菜有较大概率产量较高且保质期较短，消费快，人们对此的消费需求较频繁；洪山菜薹珍品消费频率较低，可推测其较为耐吃，可储存较长时间，人们对此的消费需求无需很频繁。

5.2 问题二模型的建立与求解

5.2.1 数据预处理

要探究销售总量 X 与售价的关系，除了二者本身，还需考虑批发价和进货量对其的影响。本文将批发价设为 x ，当日售价设为 y ，前日销售额设为 a_0 ，当日销售额设为 a 。

为了提高模型的精度，本文先对销售总量与售价进行了标准化处理，本文以对销售总量的处理为例，具体公式如下（ \bar{X} 为均值， S_X 为标准差）：

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (4)$$

$$S_X = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}} \quad (5)$$

$$X' = \frac{X - \bar{X}}{S_X} \quad (6)$$

5.2.2 销售总量与售价的回归、拟合与评估

本文将销售量设为 F ，售价设为 s ，欲使用非线性回归分析建立模型，得到形如下式的公式：

$$F = a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n \quad (7)$$

为求出系数，对曲线进行拟合，本文采用多项式拟合法对其进行估计。为在获得较为精确的函数且避免过度拟合，本文以水生根茎类为例，将拟合的多项式次数定为 6，最终所得公式如下：

$$F = -0.0575 s^6 + 4.2697 s^5 - 129.3225 s^4 + 2046.3846 s^3 - 17826.5114 s^2 + 80771.7790 s - 146677.3048 \quad (8)$$

为评估该回归模型的拟合质量，本文对其残差平方和（SSR）、总平方和（SST）进行计算，以计算得到其拟合优度。具体计算过程如下：

$$SSR = \sum_{i=1}^n (\hat{F}_i - \bar{F})^2 = 6184825.1165 \quad (9)$$

$$SST = \sum_{i=1}^n (F_i - \bar{F})^2 = 14589375.2952 \quad (10)$$

$$R^2 = 1 - \frac{SSE}{SST} = 0.4239 \quad (11)$$

综合以上分析，我们的模型在一定程度上具有良好的拟合优度，但仍有改进的空间。在实际应用中，需要根据具体情况来评估模型的适用性，并可能对模型进行进一步优化以提高其性能。

5.2.3 日平均批发价的计算、预测与检验

为探究各品类的平均批发价变化规律以进行未来一周的数据预测，本文先估计各单品的日进货量 p ，再根据各单品批发价 q 以求出该单品日批发价，最后累加各单品日批发价获得该品类日总批发价 m ，除以该品类总批发量 n 即可得该品类的日平均批发价 c 。

考虑到单品进货会有损耗，且商品存在未售出和退货情况，本文在商品固有损耗的基础上，通过查阅资料与提议对所给退货量的分析，采用 0.8 作为其系数，即认为 80% 的批发量成功售出。故单品日进货量和品类日总批发量的具体公式如下（这里以 t 表示具体单品的日销售量， β 表示该单品的损耗率）：

$$p = \frac{t}{0.8(1 - 0.01\beta)} \quad (12)$$

$$n = \sum p \quad (13)$$

该品类的日总批发价则为各单品批发价的总和：

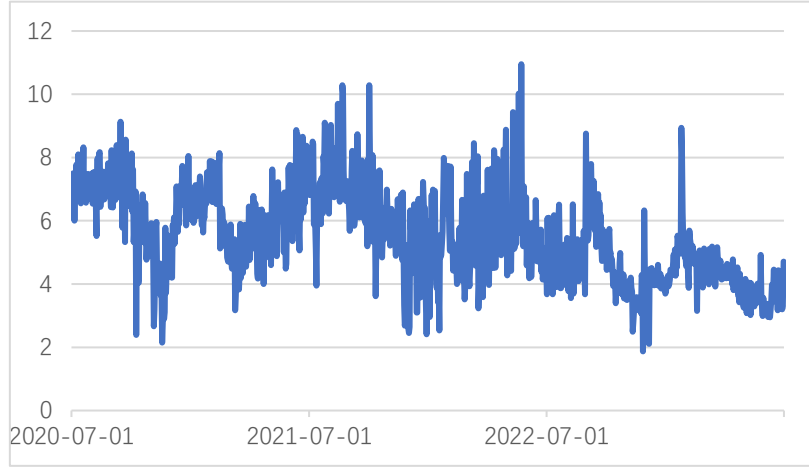
$$m = \sum pq \quad (14)$$

因此可得该品类的日平均批发价

$$c = \frac{m}{n} \quad (15)$$

算出单品的日平均批发价后，即可通过时间序列模型对未来一周的单品日批发价进行预测，本文以食用菌为例进行预测（其余预测数据可见附录）。

通过专家建模器分析，本文决定采用 ARIMA 模型对食用菌日平均批发价进行预测。品类日平均批发价作为时间序列数据，要应用 ARIMA(p,d,q) 模型，满足公式 $y_t = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \dots + \alpha_p y_{t-p} + \varepsilon_t$ 首先要对其进行平稳性检验。



图十：食用菌日平均批发价时间序列图

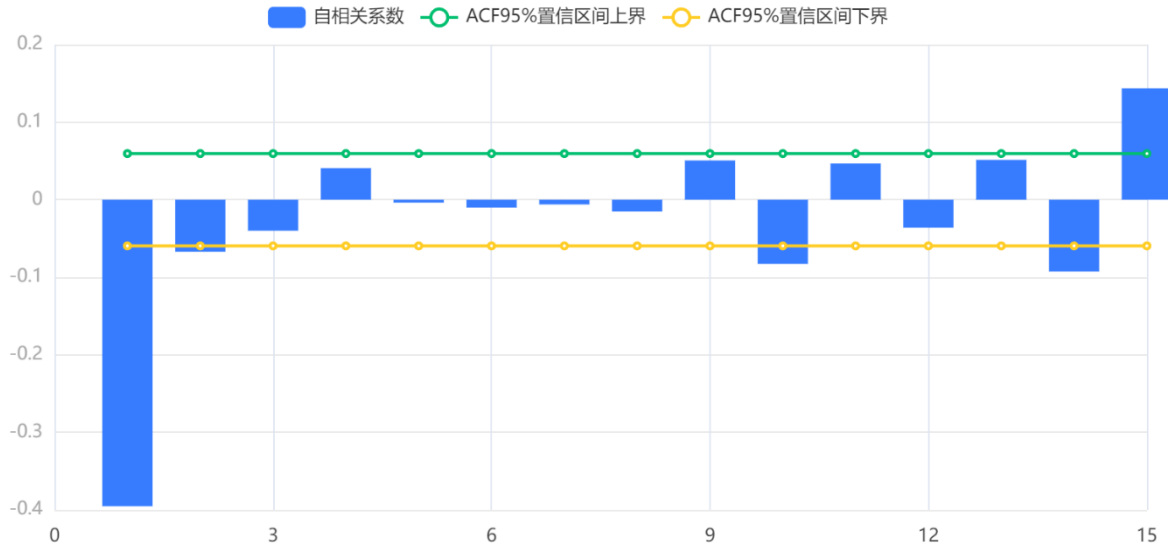
从上图发现，食用菌日平均批发价波动较大，较不平稳性，故对其用以下公式进行一阶差分（L 为滞后算子，且 $L^i y_t = y_{t-i}$ ）：

$$\Delta y_t = y_t - y_{t-1} = (1 - L)y_t \quad (16)$$

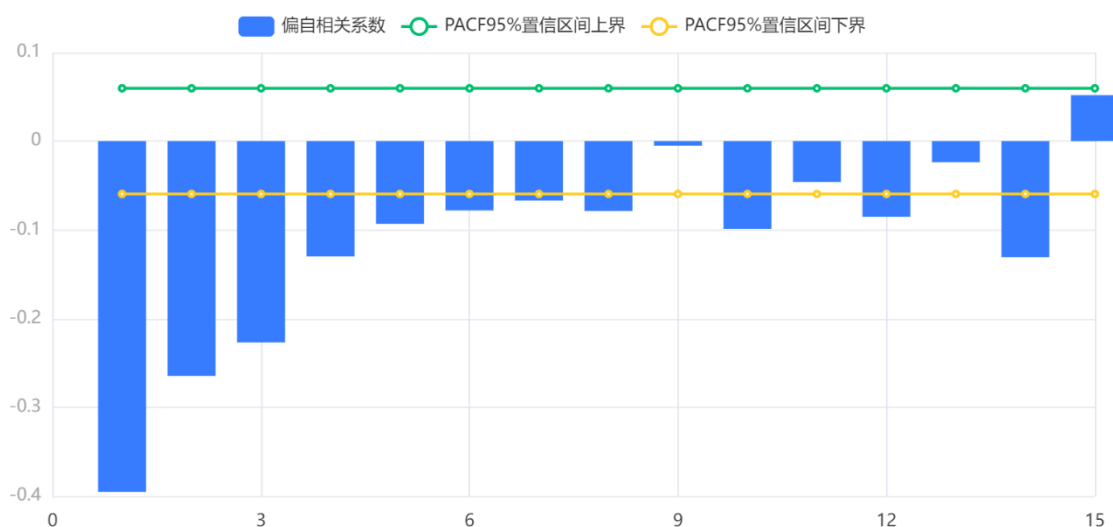
由于序列的原始数据趋势图只能大致的观测序列的平稳性，为了进一步分析其平稳性，本文用下述公式计算推出该序列的自相关系数（ACF） $\hat{\rho}$ 与偏自相关系数（PACF） $\hat{\phi}_j$ ，并最终获得下图：

$$\hat{\rho} = \frac{\sum_{t=s+1}^T (x_t - \bar{x})(x_{t-s} - \bar{x})}{\sum_{t=1}^T (x_t - \bar{x})^2} \quad (17)$$

$$x_t = \phi_{s1}x_{t-1} + \phi_{s2}x_{t-2} + \dots + \phi_{ss}x_{t-s} + \rho_t \quad (18)$$

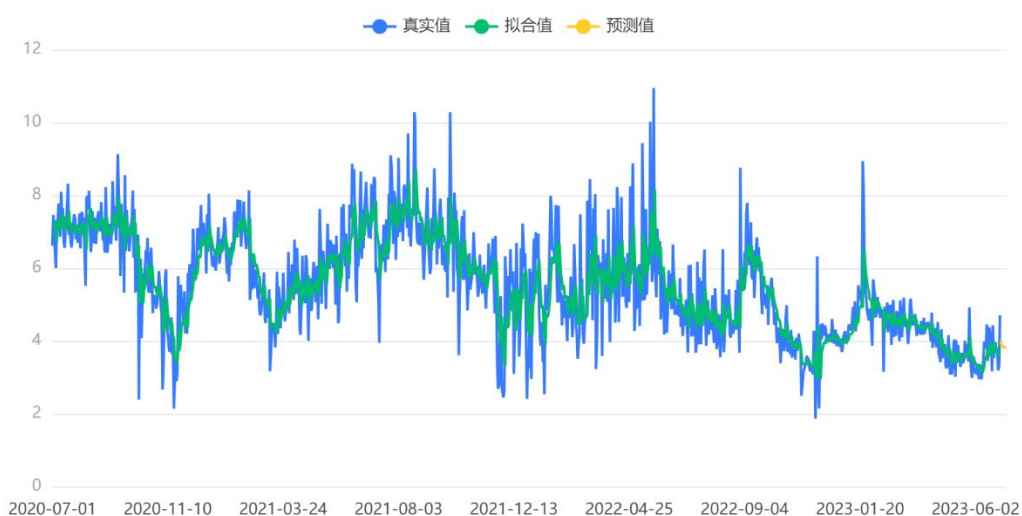


图十一：ACF 相关系数图



图十二：PACF 相关系数图

序列样本自相关系数在第一阶显著异于 0，且从图形上表现出拖尾特征，而样本 PACF 系数前五阶均显著异于 0，第九阶又趋近于 0，故选择使用 ARIMA（0，1，10）模型，最终得到如下预测：



图十三：时间序列预测图

表一：未来一周食用菌日平均批发价预测值

日期	7 月 1 日	7 月 2 日	7 月 3 日	7 月 4 日	7 月 5 日	7 月 6 日	7 月 7 日
平均批发价	4.0208	3.8252	3.8458	3.8373	3.8299	3.8354	3.8403

5.2.4 日平均损耗量计算

与平均批发价相同，先用各单品损耗量为 l 、各单品日进货量 p 计算出日总损耗量

r，再除以该品类总进货量 n 即可的该品类的日平均损耗量 d，具体公式如下：

$$r = pl \tag{19}$$

$$d = \frac{r}{n} \tag{20}$$

5. 2. 5 非线性规划求最优决策

设利润为 S，品类当日销售量为a，定价为y，批发量为 n，批发价为 c，故可写出以下目标函数：

$$S = \sum ay - nc \tag{21}$$

由于定价受批发价的影响，销售量受批发量的影响，故可列出下述两种基础约束条件：

$$y > x \tag{22}$$

$$n(1 - \beta) > a \tag{23}$$

即定价高于批发价，销售额小于考虑到损耗量后的批发量。
 本文以食用菌为例深度探究其约束条件，其余品类数据可见附录。
 通过统计分析，计算可得食用菌日销售量、日损耗率相关数据如下表：

表二：食用菌日销售量、日损耗率、日加成率、日销售额相关数据

	日销售量	日损耗率	日加成率	日销售额
平均	70.15808	7.233755	0.6172	208.3622
最小值	3.012	1.083129	0.0037	3.91385
最大值	511.136	13.12211	2.3503	2092.613

由上表可的损耗率范围，对其销售量a进行进一步约束，并对食用菌的日销售额有大概预测范围；通过加成率范围，使定价与批发价之间的条件限制更加精确，检验规划成果，在必要时刻进行调参。

同时，根据第一题所获得的各品类之间销售量的关系，通过横向对比是适当调整各品类的销售量，使数据考虑更完整 。最终获得食用菌的定价与补货量如下表所示：

表三：未来一周食用菌定价与补货量规划

食用菌	预测批发价	定价	补货量（千克）	加成率	最大利润
7月1日	4.0208378	7.79	30	0.937407	618.43
7月2日	3.8251632	7.26	30	0.897958	609.51
7月3日	3.8457598	6.95	30	0.807185	594.19
7月4日	3.8372727	7.11	30	0.852878	603.02
7月5日	3.8298941	7.21	30	0.882559	607.55
7月6日	3.8353824	7.13	30	0.859006	604.27
7月7日	3.8402742	7.06	30	0.83841	600.55
总计					4237.52

整理各品类相关数据，最终获得未来一周各品类定价与补货量的情况如下表：

表四：未来一周各品类补货量的情况

花菜类	花叶类	辣椒类	茄类	食用菌	水生根茎类
109.9	60	50	75	30	25
110	60	50	75	30	25
110	60	50	75	30	25
110	60	50	75	30	25
110	60	50	75	30	25
110	60	50	75	30	25
110	60	50	75	30	25

表五：未来一周各品类定价的情况

	花菜类	花叶类	辣椒类	茄类	食用菌	水生根茎类
7月1日	10.71	4.15	9.99	5.45	7.79	14.99
7月2日	10.71	4.12	9.99	5.45	7.26	14.99
7月3日	10.71	4.09	9.99	5.45	6.95	14.99
7月4日	10.71	4.06	9.99	5.45	7.11	14.99
7月5日	10.71	4.04	9.99	5.45	7.21	14.99
7月6日	10.71	4.02	9.99	5.45	7.13	14.99
7月7日	10.71	4	9.99	5.45	7.06	14.99

表六：未来一周每日利润

日期	最大利润	日期	最大利润
7月1日	2599.92	7月5日	2633.588
7月2日	2600.5186	7月6日	2641.093
7月3日	2598.1381	7月7日	2648.203
7月4日	2617.6881		
总计	10416.2648		

5.3 问题三模型的建立与求解

5.3.1 数据预处理

本文首先对6月23日至6月30日的数据进行了数据预处理，将数据按照品类分为六组并分开进行数据分析、数据清洗，最后得到了50组不同单品的汇总图，其中横向表示的是每一个单品的总利润、日利润、退货率、销量、购买频率，这些数据直接反映该单品蔬菜在市场上的受欢迎程度以及能够带来的经济效应。日利润、总利润、销量直接反映该蔬菜的重要性。故经过综合排序后取重要性前33名的蔬菜作为数据筛选，下图反映的前9名的蔬菜以及它们的日利润（其余可见附录3）：

表七：单品及其日利润

单品类型	日利润
西兰花	0.44172864
云南生菜(份)	0.88570796
净藕(1)	0.46871111
小米椒(份)	3.62033333

云南油麦菜(份)	1.2895302
金针菇(盒)	0.42628319
芜湖青椒(1)	0.67155522
枝江青梗散花	0.49417647
竹叶菜	0.62219488

5.3.2 粒子群算法模型建立

粒子群算法模型（Particle Swarm Optimization）的建立，粒子群算法模型是基于初始化粒子群，并且根据特定的目标函数粒子不断更新个体以及全局最佳位置，经过不断迭代和更新速度和位置，最后得到目标函数的最大值或者最小值。本文建立的利润目标函数如下：

$$\text{利润} = \text{销量} \times (\text{销售价} - \text{批发价}) \quad (24)$$

决策变量为各个单品的补货总量和选择的单品数量，对于补货总量，需要预估 7 月 1 日用户对所有蔬菜单品的需求量，故通过线性回归模型拟合出 7 月 1 日的销量、销售价格以及批发价格，其中销量一定程度上反映了用户对不同蔬菜单品的需求量，故补货总量需要衡量预测出的用户需求量以及批发价格。将得到的预测销量、预测批发价格、预测销售价格代入粒子群算法的参数中，可以得到生成的目标函数利润的最大值，同时也能够满足用户对蔬菜的大量需求。7 月 1 日预测出的单 49 品蔬菜对应的批发价、销售价格和销量见附录 5

粒子群算法最终的到了 7 月 1 日不同单品它们所需要的最适合的补货总量以及定价策略，其中，西兰花的最适合的补货总量，最适合的销售定价为 12.56 元，最适合的单品补货量为 12.15kg，利润为 56.43 元，最终汇总的到总收益为 750 元。

5.4 问题四的模型建立

要更好地制定蔬菜商品地补货和定价决策，首先确定商超的主要目标：增加总利润以及建立良好信誉。为此，本文建议额外采取周围商超位置客源及蔬菜商品种类供应量、该商超的客户群体种类数量信息数据、该商超地蔬菜库存实时监测数据、批发地的距离品种数量品质数据、相关蔬菜商品的网络时事热点趋势动向数据五类数据。

5.4.1 周围商超位置客源及蔬菜商品种类供应量的数据

市场客源是商超经营的根本经济来源，若附近存在同类型商超，会出现目标客户重叠的情况，争抢市场，降低蔬菜商品的销售量，造成库存堆积，累积亏损。持续关注周围商超的经营状况及蔬菜种类和供应量，可以及时做出相应的对策，调整自身进货量及种类：提前引进季节时令性蔬菜，占领当季蔬菜市场，若附近商超已抢占某蔬菜商品的市场，还可错开该商品类型，避免无效购入积压库存造成亏损，还可共同满足更加全面的客户群体需求，提升当地客户消费满意度；激励自身寻找更低价优质商品，提升自身商品竞争力，规避相互盲目攀比降价。

5.4.2 该商超的客户群体种类数量信息数据

了解该商超的服务群体以及客户需求，才能较完善地制定进货种类数量和定价的策略。本文以旧城镇居民区、商业区、新兴发达城市为例进行分析。

本文将蔬菜按照其需求程度进行分类：

A.大量必需的绿色蔬菜

a 全年型蔬菜

b 季节性蔬菜

c 外地进口蔬菜

B.需求较次的茄类辣椒类

C.需求最小的特产补品礼盒类

若该商超靠近旧城镇居民区，大部分消费人群可能为中老年人，以满足近日的自身三餐饮食的蔬菜需求为主要消费目标。A 与 B 的购买需求较大，且多为散装小批量销售，小程度的优惠售价，例如捆绑购买降价或附送少量例如葱姜蒜的调味品，通过充值优惠等政策可以较为有效地吸引和固定客户。而 C 类商品对该类人群的吸引力度并不高，可以维持在一个较低保有量，于特定节假日前适量购入，以应对该时期的特产礼盒赠送需求。

若位于商业区，消费群体则偏向于餐饮行业商铺，其购买行为偏向于大规模多种类且固定的蔬菜商品，以匹配其固定菜式菜单和较大规模出餐。该种情况的销售模式可以大批量批发为主，并签订建立长期供应关系协议，其单位售价会根据批发量和签订协议时间长短进行一定程度的调整，从而建立较为稳定的进货量与定价。

若地处相对一线的新兴发达城市，年轻人群体的占比较大，其主要特征是对售价浮动的敏感程度较小，对新商品的好奇程度大，购买欲强，其消费潜力相较于中老年人更大，且对蔬菜商品的预处理程度要求更高。针对这类人群，不必刻意降低售价，相比于自由散装蔬菜商品，更好的销售方式是销售已完成清洗加工的少量分装蔬菜，这种小包装商品更利于售卖，且满足年轻群体更看重效率和便捷性的生活需要。

5.4.3 该商超的蔬菜库存实时监测数据

蔬菜库存的监控数据是蔬菜进货量的重要影响因素之一，这直接体现了蔬菜商品的每日销售情况，也可根据该数据来推测人群的消费意愿，对进货量进行及时的调整。若库存已有一定基础时仍照常进货，则易造成库存堆积，累积亏损。此外，对蔬菜库存进行实时监控，可对其制定销售方案及时进行打折促销处理，最大程度上减少因蔬菜品质降低而滞销的浪费现象，从产品链尾部优化方案。

5.4.4 批发地的距离品种数量品质数据

在完成市场需求调研后，就可以根据需求对批发商品来源进行筛选分析，批发地的供货信息包括产地距离，运输方式及运费，是否需要冷藏及保鲜费用，商品的品种数量，对目标批发商品的供应能力（能否在特定时间及时供应满足需求量），商品质量，批发价格，能否长期稳定供应蔬菜商品等信息。对以上指标进行综合评判才能筛选确定最理想的供货方案，在货源把握最优决策。

5.4.5 相关蔬菜商品的网络时事热点趋势动向数据

关注网络的相关热点信息，有利于及时预测并把握最近的消费高峰和低谷，在消费热潮前及时增加供应，适当调整售价和销售模式，发现某蔬菜出现负面消息，或者可能因自然灾害导致蔬菜供应不稳或者品质下降前，及时减少批发，可以有效规避风险，把握市场规律，紧跟时事调整补货量与定价，增加利润。

六、模型的分析与检验

6.1 灵敏度分析

为了评价模型的灵敏度，我们需要关注模型对自变量（售价与销售额）变化的敏感程度。首先我们定义自变量的变化范围为 $[0,1]$ ，并计算出原始水生根茎类售价与销售额逻辑回归模型的准确率为 38.24%。我们将每个自变量在设定变化范围内进行变化，并获取自变量变化后的模型预测结果。经观察可得，模型呈现出 42.39% 的准确率，这表明自变量的变化会导致准确率的偏差，即我们的模型对数据的敏感性较强，当面对数据的偏差时，准确性会受较大影响。同时我们对灵敏度分析的灵敏度输出进行检测，正的灵敏度表示增加自变量会导致对模型的预测结果产生正面影响，而负的灵敏度表示减少自变量会导致预测结果产生负面影响。

七、模型的评价

7.1 模型的优点

本文使用了数据可视化、皮尔逊相关系数分析、非线性回归、多项式拟合、非线性规划、ARIMA 时间序列预测、粒子群算法模型等模型，根据各个模型和方法的表现与输出结果，我们总结出了以下各个模型的优点。

- 除了对各品类、各单品的数据进行横向对比分析，还时间按照季度、月份、每日分类对各品类、单品进行了纵向分析，整体比例与个体趋势分析较为全面。
- 在建模之前均进行了数据预处理，填补缺失值、剔除异常值、标准化数据，提高迭代求解精度，提升模型的收敛速度，使所得成果更加真实可靠。
- 大部分模型经过检验与调整，除了对每个模型成立的前提检验，也对时间序列模型、非线性回归模型的预测准确率进行检验，并根据结果对原模型进行了优化，使数据更贴合实际。
- 模型具有较高的实际意义，计算较为严谨，在非线性规划中，通过计算历年日平均数据和最值增加限制条件，精确自变量范围，将理想中的数据与现实数据相结合，使模型更具有客观性。
- 在解决问题时，应用多个模型组合，避免了单一模型建模的局限性，最大程度的提高文章的丰富性，满足题目多种限制的需要，使解答更加充实完善。

7.2 模型的缺点

- 未对单品之间的相互关系进行深刻分析，未探究不同品类单品之间可能存在的捆绑销售、互替等关系。
- 未进行各品类的平均损耗率的动态调整，预测当日销售额与当日拟定价的回归方程拟合度不高。
- 预测时未进行各品类平均损耗量和销售进货占比参数的每日动态调整。

八、参考文献

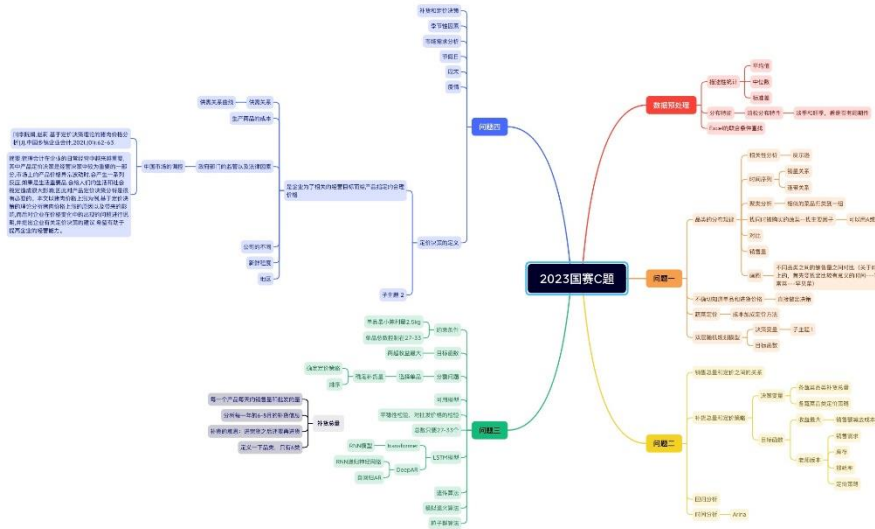
- [1] 毛莉莎, 供应链视角下蔬菜批发市场定价策略及产销模式研究, 2023
- [2] 张鹤, 基于集成学习的蔬菜销售预测——以昆明某超市为例, 2021
- [3] Janda Kathryn M.Ranit Nalin; Salvo Deborah; Nielsen Aida; kaliszewski Catherine; Hoe'scher Deanna M.van den Bero Alexandra E., Association between Fresh Fruit and Vegetable Consumption and Purchasing Behaviors, Food InsecurityStatus and Geographic Food Access among a Lower-Income, Racially/Ethnically Diverse Cohort in CentraTexas, Volume 14 (23), 2022, PP. 5149-5149
- [4] 李阮娟, 赵莉. 基于定价决策理论的猪肉价格分析[J]. 中国乡镇企业会计, 2021, (01):62-63.

附录

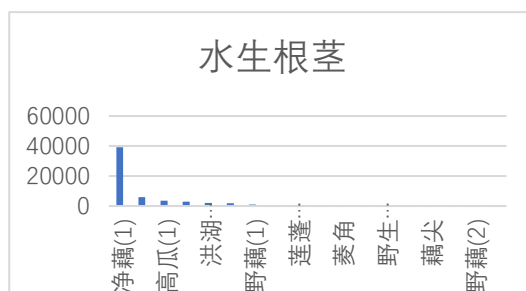
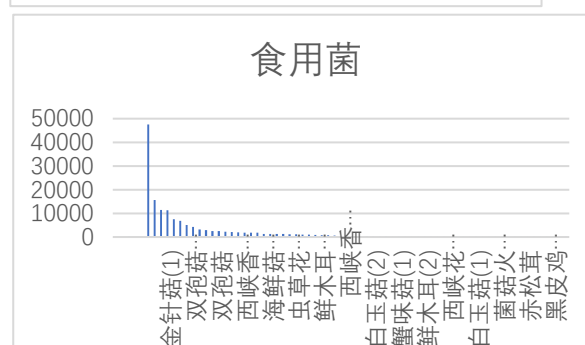
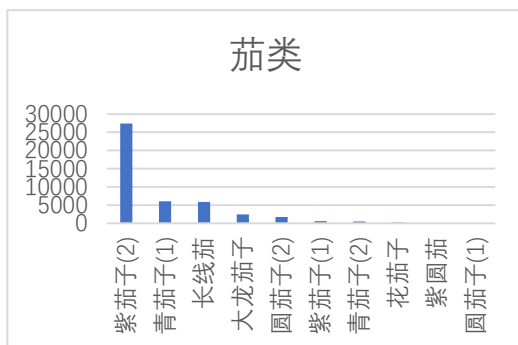
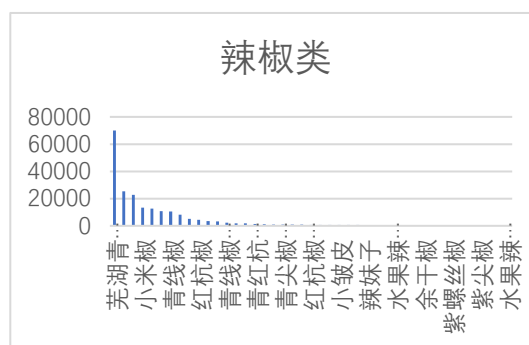
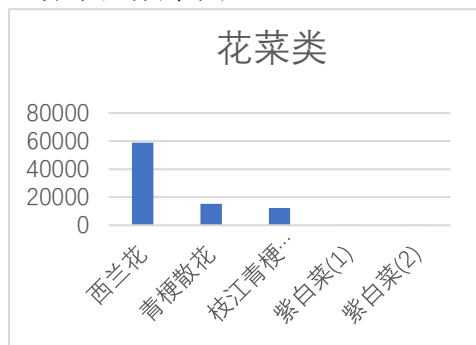
附录 1

介绍：支撑材料的文件列表

总体思路分析



各单品频率图：



附录 2

介绍：

问题二各品类回归拟合与规划的代码

ARIMA 模型预测代码：

```
% 1. 读取 Excel 文件中的数据
```

```
% 假设您的 Excel 文件名为'食用菌.xlsx'，第一列为日期，第二列为平均批发价。
```

```
% 读取 Excel 文件
```

```
filename = '食用菌.xlsx';
```

```
sheet = 1; % 假设数据在第一个工作表中
```

```

data = xlsread(filename, sheet);

% 提取日期和平均批发价列
dates = data(:, 1);
prices = data(:, 2);

% 2. 数据预处理
% a. 检查数据是否平稳
[h, pValue, stat, cValue, reg] = adftest(prices);
if h == 0
    disp('时间序列是平稳的。');
else
    disp('时间序列不是平稳的。');
end

% b. 进行差分操作，直到数据变得平稳
d = 0;
while h ~= 0
    prices = diff(prices);
    d = d + 1;
    [h, pValue, stat, cValue, reg] = adftest(prices);
end
disp(['进行了 ' num2str(d) ' 次差分操作，使数据平稳。']);

```

```

% 3. 建立 ARIMA 模型

p = 1; % 自回归阶数

q = 1; % 移动平均阶数

model =
arima('ARLags',1:p,'MALags',1:q,'D',d,'Distribution','Gaussian');

% 4. 估计模型参数

estModel = estimate(model, prices);

% 5. 模型预测

forecastHorizon = 7; % 预测的天数

lastObservedPrice = prices(end); % 获取最后一个观测值

[forecastedPrices, forecastedPriceSE] = forecast(estModel,
forecastHorizon, 'Y0', lastObservedPrice);

% 6. 显示预测结果

disp('预测结果:');

disp(forecastedPrices);

% 可以将预测结果保存到 Excel 文件中，以便进一步分析或可视化。

% 例如，将日期和预测结果合并为一个表格，并保存到新的 Excel 文件：

forecastTable = table(dates(end), forecastedPrices, 'VariableNames',
{'Date', 'ForecastedPrice'});

writetable(forecastTable, '预测结果.xlsx');

```

花菜类回归拟合的代码：

```

clear;clc
% 定义多项式系数 a
a = [-0.4422, 22.4808, -452.9520, 4502.7153, -
22061.8909, 43863.0541];
% 定义常数 b
b = 7.7141;
% 定义目标函数
fun = @(vars) -
(vars(1) * polyval(a, vars(1)) / 8.5 - vars(2) * b);
% 初始猜测值
x0 = [9, 100]; % 选择一个合适的初始值 [x, n]
% 定义非线性不等式约束和非线性等式约束
nonlcon = @(vars) constraint(vars, a, b);
% 设置变量的上下界
lb = [b + eps, 20]; % x 大于 b, n 大于 20
ub = [11, 200]; % x 小于等于 11, n 小于等于 200
% 调用 fmincon 求解
options = optimoptions('fmincon', 'Display', 'iter'); % 设置
显示迭代过程
[x_opt, fval] = fmincon(fun, x0, [], [], [], [], lb, ub,
nonlcon, options);
% 输出结果
x_optimal = x_opt(1);
n_optimal = x_opt(2);
s_max = -fval;
disp(['最优解 x = ', num2str(x_optimal)]);
disp(['最优解 n = ', num2str(n_optimal)]);
disp(['目标函数最大值 s = ', num2str(s_max)]);
% 定义非线性约束函数
function [c, ceq] = constraint(vars, a, b)
    x = vars(1);
    n = vars(2);
    c = [0.8 * n * (1 - 0.105453) - polyval(a, x);
% 不等式约束
                                n - 20; % 不等式约束
                                200 - n]; % 不等式约束
    ceq = []; % 等式约束为空
end
花菜类规划的代码：
function [result, ranks, rank_sum_ratio] = rank_sum_evaluation
(data, weights)
% data: 数据矩阵，每一行代表一个样本，每一列代表一个评价指标
% weights: 各评价指标的权重向量，长度与评价指标个数相同

```

```

% 标准化数据
normalized_data = normalize(data);

% 计算加权得分
weighted_scores = normalized_data * weights';

% 计算秩和比
ranks = tiedrank(weighted_scores);
rank_sum = sum(ranks);
rank_sum_ratio = rank_sum / (size(data, 1) * (size(d
ata, 1) + 1) / 2);

% 输出结果
result = rank_sum_ratio;

% 可选：输出每个样本的加权得分和秩
disp("加权得分和秩：");
disp([weighted_scores, ranks]);

% 可选：绘制加权得分和秩的图表
figure;
subplot(2, 1, 1);
bar(weighted_scores);
xlabel('样本');
ylabel('加权得分');
title('加权得分图');

subplot(2, 1, 2);
bar(ranks);
xlabel('样本');
ylabel('秩');
title('秩图');
end
花叶类回归拟合代码：
clear;clc
% 定义多项式系数 a

```



```

a = [1.7960, -165.4095, 3067.3427, -
22734.5044, 73344.1837, -78821.7212];
% 定义常数 b
b = 2.983
;
% 定义目标函数
fun = @(vars) -
(vars(1) * polyval(a, vars(1))/30 - vars(2) * b);
% 初始猜测值
x0 = [5, 100]; % 选择一个合适的初始值 [x, n]
% 定义非线性不等式约束和非线性等式约束
nonlcon = @(vars) constraint(vars, a, b);
% 设置变量的上下界
lb = [b + eps, 20]; % x 大于 b, n 大于 200
ub = [6, 60]; % x 小于等于 6, n 小于等于 800
% 调用 fmincon 求解
options = optimoptions('fmincon', 'Display', 'iter'); % 设置
显示迭代过程
[x_opt, fval] = fmincon(fun, x0, [], [], [], [], lb, ub,
nonlcon, options);
% 输出结果
x_optimal = x_opt(1);
n_optimal = x_opt(2);
s_max = -fval;
disp(['最优解 x = ', num2str(x_optimal)]);
disp(['最优解 n = ', num2str(n_optimal)]);
disp(['目标函数最大值 s = ', num2str(s_max)]);
% 定义非线性约束函数
function [c, ceq] = constraint(vars, a, b)
x = vars(1);
n = vars(2);
c = [0.8 * n * (1 - 0.124426) - polyval(a, x);
% 不等式约束
n - 200; % 不等式约束
800 - n]; % 不等式约束
ceq = []; % 等式约束为空
end
花叶类规划代码
function [result, ranks, rank_sum_ratio] = rank_sum_evaluation
(data, weights)
% data: 数据矩阵, 每一行代表一个样本, 每一列代表一个评价指标
% weights: 各评价指标的权重向量, 长度与评价指标个数相同

```

```

% 标准化数据
normalized_data = normalize(data);

% 计算加权得分
weighted_scores = normalized_data * weights';

% 计算秩和比
ranks = tiedrank(weighted_scores);
rank_sum = sum(ranks);
rank_sum_ratio = rank_sum / (size(data, 1) * (size(d
ata, 1) + 1) / 2);

% 输出结果
result = rank_sum_ratio;

% 可选：输出每个样本的加权得分和秩
disp("加权得分和秩：");
disp([weighted_scores, ranks]);

% 可选：绘制加权得分和秩的图表
figure;
subplot(2, 1, 1);
bar(weighted_scores);
xlabel('样本');
ylabel('加权得分');
title('加权得分图');

subplot(2, 1, 2);
bar(ranks);
xlabel('样本');
ylabel('秩');
title('秩图');
end
辣椒类回归拟合代码：
clear;clc
% 导入 Excel 数据
data = xlsread('辣椒类.xlsx'); % 假设数据存储在名为“食用菌数
据.xlsx”的文件中

```

```

% 列索引
sales_column = 1; % 销售额列
price_column = 2; % 售价列
% 步骤 1: 检查异常值
% 使用箱线图检测异常值
figure;
boxplot(data(:, sales_column));
title('销售额箱线图 - 检测异常值');
% 根据箱线图上的异常值范围进行异常值剔除
sales_data = data(data(:, sales_column) <= quantile(data(:, sales_column), 0.75) + 1.5 * iqr(data(:, sales_column)) & data(:, sales_column) >= quantile(data(:, sales_column), 0.25) - 1.5 * iqr(data(:, sales_column)), :);
% 步骤 2: 数据平滑 (这里采用移动平均作为示例)
window_size = 3; % 平滑窗口大小
smoothed_sales_data = movmean(sales_data(:, sales_column), window_size);
% 步骤 3: 数据清理后的效果
figure;
subplot(2, 1, 1);
plot(data(:, sales_column), 'b-o');
title('原始销售额数据');
subplot(2, 1, 2);
plot(smoothed_sales_data, 'g-o');
title('经过异常值检测、平滑和清理后的销售额数据');
% 显示结果
fprintf('原始销售额数据长度: %d\n', size(data, 1));
fprintf('处理后销售额数据长度: %d\n', length(smoothed_sales_data));
% 提取销售额和售价数据
sales = sales_data(:, 1); % 第一列是销售额
price = sales_data(:, 2); % 第二列是售价
% 设置多项式的次数
degree = 6; % 可根据需要调整多项式的次数
% 使用 polyfit 进行多项式拟合
coefficients = polyfit(price, sales, degree);
% 生成拟合曲线的数据点
fitPrice = linspace(min(price), max(price), 100)'; % 在售价范围内生成 100 个数据点
fitSales = polyval(coefficients, fitPrice);
% 绘制散点图和拟合曲线
figure;
scatter(price, sales, 'b', 'filled'); % 绘制散点图
hold on;
plot(fitPrice, fitSales, 'r', 'LineWidth', 2); % 绘制拟合曲线

```

```

xlabel(' 售价 ');
ylabel(' 销售额 ');
title(' 销售额与售价的多项式拟合 ');
legend(' 数据点', ' 拟合曲线 ');
% 计算拟合优度 (R-squared)
predictedSales = polyval(coefficients, price); % 预测的销售额
sse = sum((predictedSales - sales).^2); % 残差平方和
sst = sum((sales - mean(sales)).^2); % 总平方和
r_squared = 1 - sse / sst; % 拟合优度
fprintf(' 拟合优度 (R-squared): %.4f\n', r_squared);
% 显示拟合多项式的表达式
fprintf(' 拟合多项式表达式: ');
for i = degree:-1:0
    fprintf('%.4f * x^%d', coefficients(degree - i + 1),
i);
    if i > 0
        fprintf(' + ');
    end
end
fprintf('\n');
辣椒类规划代码:
clear;clc
% 定义多项式系数 a
a = [-0.0442, 2.8956, -75.6103, 1002.4238, -
7050.6940, 24405.6430, -28589.7058];
% 定义常数 b
b = 4.3771;
% 定义目标函数
fun = @(vars) -
(vars(1) * polyval(a, vars(1)) / 30 - vars(2) * b);
% 初始猜测值
x0 = [9, 100]; % 选择一个合适的初始值 [x, n]
% 定义非线性不等式约束和非线性等式约束
nonlcon = @(vars) constraint(vars, a, b);
% 设置变量的上下界
lb = [b + eps, 10]; % x 大于 b, n 大于 10
ub = [10, 90]; % x 小于等于 10, n 小于等于 90
% 调用 fmincon 求解
options = optimoptions('fmincon', 'Display', 'iter'); % 设置
显示迭代过程
[x_opt, fval] = fmincon(fun, x0, [], [], [], [], lb, ub,
nonlcon, options);
% 输出结果
x_optimal = x_opt(1);

```

```

n_optimal = x_opt(2);
s_max = -fval;
disp(['最优解 x = ', num2str(x_optimal)]);
disp(['最优解 n = ', num2str(n_optimal)]);
disp(['目标函数最大值 s = ', num2str(s_max)]);
% 定义非线性约束函数
function [c, ceq] = constraint(vars, a, b)
    x = vars(1);
    n = vars(2);
    c = [0.8 * n * (1 - 0.063146) - polyval(a, x);
% 不等式约束
                                n - 10; % 不等式约束
                                90 - n]; % 不等式约束
    ceq = []; % 等式约束为空
end
茄类回归拟合代码：
clear;clc
% 导入 Excel 数据
data = xlsread('茄类.xlsx'); % 假设数据存储在名为“食用菌数
据.xlsx”的文件中
% 列索引
sales_column = 1; % 销售额列
price_column = 2; % 售价列
% 步骤 1: 检查异常值
% 使用箱线图检测异常值
figure;
boxplot(data(:, sales_column));
title('销售额箱线图 - 检测异常值');
% 根据箱线图上的异常值范围进行异常值剔除
sales_data = data(data(:, sales_column) <= quantile(data(:, s
ales_column), 0.75) + 1.5 * iqr(data(:, sales_column)) & data(:,
sales_column) >= quantile(data(:, sales_column), 0.25) - 1.5
iqr(data(:, sales_column)), :);
% 步骤 2: 数据平滑（这里采用移动平均作为示例）
window_size = 3; % 平滑窗口大小
smoothed_sales_data = movmean(sales_data(:, sales_column), wind
ow_size);
% 步骤 3: 数据清理后的效果
figure;
subplot(2, 1, 1);
plot(data(:, sales_column), 'b-o');
title('原始销售额数据');
subplot(2, 1, 2);
plot(smoothed_sales_data, 'g-o');

```

```

title('经过异常值检测、平滑和清理后的销售额数据');
% 显示结果
fprintf('原始销售额数据长度: %d\n', size(data, 1));
fprintf('处理后销售额数据长度: %d\n', length(smoothed_sales_data));
% 提取销售额和售价数据
sales = sales_data(:, 1); % 第一列是销售额
price = sales_data(:, 2); % 第二列是售价
% 设置多项式的次数
degree = 9; % 可根据需要调整多项式的次数
% 使用 polyfit 进行多项式拟合
coefficients = polyfit(price, sales, degree);
% 生成拟合曲线的数据点
fitPrice = linspace(min(price), max(price), 100)'; % 在售价范围
% 围内生成 100 个数据点
fitSales = polyval(coefficients, fitPrice);
% 绘制散点图和拟合曲线
figure;
scatter(price, sales, 'b', 'filled'); % 绘制散点图
hold on;
plot(fitPrice, fitSales, 'r', 'LineWidth', 2); % 绘制拟合曲线
xlabel('售价');
ylabel('销售额');
title('销售额与售价的多项式拟合');
legend('数据点', '拟合曲线');
% 计算拟合优度 (R-squared)
predictedSales = polyval(coefficients, price); % 预测的销售额
sse = sum((predictedSales - sales).^2); % 残差平方和
sst = sum((sales - mean(sales)).^2); % 总平方和
r_squared = 1 - sse / sst; % 拟合优度
fprintf('拟合优度 (R-squared): %.4f\n', r_squared);
% 显示拟合多项式的表达式
fprintf('拟合多项式表达式: ');
for i = degree:-1:0
    fprintf('%.4f * x^%d', coefficients(degree - i + 1),
i);
    if i > 0
        fprintf(' + ');
    end
end
fprintf('\n');
茄类规划模型:
clear;clc
clear;clc
% 定义多项式系数 a

```

```

a = [0.0410, -3.5794, 137.1573, -3026.9513, 42372.8815, -
389919.3946, 2357168.1605, -9021735.9240, 19827199.8729, -
19055396.0690];
% 定义常数 b
b = 4.5328;
% 定义目标函数
fun = @(vars) -
(vars(1) * polyval(a, vars(1)) / 10 - vars(2) * b);
% 初始猜测值
x0 = [6, 70]; % 选择一个合适的初始值 [x, n]
% 定义非线性不等式约束和非线性等式约束
nonlcon = @(vars) constraint(vars, a, b);
% 设置变量的上下界
lb = [b + eps, 10]; % x 大于 b, n 大于 10
ub = [10, 14]; % x 小于等于 10, n 小于等于 140
% 调用 fmincon 求解
options = optimoptions('fmincon', 'Display', 'iter'); % 设置
显示迭代过程
[x_opt, fval] = fmincon(fun, x0, [], [], [], [], lb, ub,
nonlcon, options);
% 输出结果
x_optimal = x_opt(1);
n_optimal = x_opt(2);
s_max = -fval;
disp(['最优解 x = ', num2str(x_optimal)]);
disp(['最优解 n = ', num2str(n_optimal)]);
disp(['目标函数最大值 s = ', num2str(s_max)]);
% 定义非线性约束函数
function [c, ceq] = constraint(vars, a, b)
x = vars(1);
n = vars(2);
c = [0.8 * n * (1 - 0.063146) - polyval(a, x);
% 不等式约束
n - 10; % 不等式约束
140 - n]; % 不等式约束
ceq = []; % 等式约束为空
end
食用菌回归拟合代码:
clear;clc
% 导入数据
data = xlsread('食用菌.xlsx');
sales = data(:, 1); % 销售量
price = data(:, 2); % 售价

```

```

% 数据处理 - 如果需要, 你可以在这里进行数据处理, 例如移除异常值或
% 标准化数据。
% 建立回归模型
mdl = fitlm(price, sales);
% 输出回归模型摘要
disp('回归模型摘要:');
disp(mdl);
% 计算 R-squared 值
rsquared = mdl.Rsquared.Ordinary;
disp(['拟合程度 (R-squared): ', num2str(rsquared)]);
% 获取回归方程参数
coefficients = mdl.Coefficients.Estimate;
intercept = coefficients(1);
slope = coefficients(2);
% 绘制散点图和回归线
figure;
scatter(price, sales, 'filled');
hold on;
x_fit = linspace(min(price), max(price), 100);
y_fit = intercept + slope * x_fit;
plot(x_fit, y_fit, 'r--', 'LineWidth', 2);
xlabel('售价');
ylabel('销售量');
title('销售量与售价关系');
legend('数据点', '回归线');
% 显示回归方程
equation = ['回 归 方 程 : 销 售',
量 = ', num2str(intercept), ' + ', num2str(slope), ' * 售 价',
'];
text(0.6, 0.1, equation, 'Units', 'normalized', 'FontSize', 1
2, 'Color', 'blue');
% 保存图像
saveas(gcf, '销售量与售价关系.png');
% 显示图像
hold off;
食用菌规划代码:
clear;clc
% 定义多项式系数 p
p = [26.5187, -705.7989, 5855.2803, -12874.4298];
% 定义常数 b
b = 3.840274243;
% 定义目标函数
fun = @(vars) -
(vars(1) * polyval(p, vars(1))/50 - vars(2) * b);

```



```

% 初始猜测值
x0 = [10, 500]; % 选择一个合适的初始值 [x, n]
% 定义非线性不等式约束和非线性等式约束
nonlcon = @(vars) constraint(vars, p, b);
% 设置变量的上下界
lb = [b + eps, 30]; % x 大于 b, n 大于 3
ub = [15, 1000]; % x 小于等于 15, n 小于等于 1000
% 调用 fmincon 求解
options = optimoptions('fmincon', 'Display', 'iter'); % 设置
显示迭代过程
[x_opt, fval] = fmincon(fun, x0, [], [], [], [], lb, ub,
nonlcon, options);
% 输出结果
x_optimal = x_opt(1);
n_optimal = x_opt(2);
s_max = -fval;
disp(['最优解 x = ', num2str(x_optimal)]);
disp(['最优解 n = ', num2str(n_optimal)]);
disp(['目标函数最大值 s = ', num2str(s_max)]);
% 定义非线性约束函数
function [c, ceq] = constraint(vars, p, b)
    x = vars(1);
    n = vars(2);
    a = polyval(p, x);
    c = [0.8 * n * (1 - 0.072331) - a; % 不等式约束
         n - 3]; % 不等式约束
    ceq = []; % 等式约束为空
end
数据预处理
% 导入 Excel 数据
data = xlsread('食用菌.xlsx'); % 假设数据存储在名为“食用菌数
据.xlsx”的文件中
% 列索引
sales_column = 1; % 销售额列
price_column = 2; % 售价列
% 步骤 1: 检查异常值
% 使用箱线图检测异常值
figure;
boxplot(data(:, sales_column));
title('销售额箱线图 - 检测异常值');
% 根据箱线图上的异常值范围进行异常值剔除
sales_data = data(data(:, sales_column) <= quantile(data(:, s
ales_column), 0.75) + 1.5 * iqr(data(:, sales_column)) & data(

```

```

, sales_column) >= quantile(data(:, sales_column), 0.25) - 1.5
* iqr(data(:, sales_column)), :);
    % 步骤 2: 数据平滑 (这里采用移动平均作为示例)
    window_size = 3; % 平滑窗口大小
    smoothed_sales_data = movmean(sales_data(:, sales_column), window_size);
    % 步骤 3: 数据清理后的效果
    figure;
    subplot(2, 1, 1);
    plot(data(:, sales_column), 'b-o');
    title('原始销售额数据');
    subplot(2, 1, 2);
    plot(smoothed_sales_data, 'g-o');
    title('经过异常值检测、平滑和清理后的销售额数据');
    % 显示结果
    fprintf('原始销售额数据长度: %d\n', size(data, 1));
    fprintf('处理后销售额数据长度: %d\n', length(smoothed_sales_data))

```

附录 3 particles.m

介绍: 粒子群算法分析, 建立规划模型求利润最大值

```

% 预估的批发价平均值
wholesale_prices = [
    3.5025
    2.66625
    1.953387097
    1.453362832
    3.402571429
    15.6
    1.3
    3.25
    4.082258065
    3.8775
    6.984931507
    4.34
    3.737021277
    9.635294118
    4.134897959
    4.606
    3.190740741
    3.222735849

```

```
1.566666667
2.5314
4.131969697
4.757260274
8.241428571
2.325583333
2.842222222
5.741666667
3.57579646
4.34
2.869530201
2.330139535
7.772964824
8.945490196
12.97323077
2.45877551
7.481653543
3.32835443
12.27466667
3.210666667
2.83
3.367573529
2.149
1.523417722
11.55
13.46478261
5.588333333
18
10.734
9.168695652
16.06666667
];
```

```
% 预估的销售平均价格
```

```
retail_prices = [  
    5.7  
    3.6125  
    2.748387097  
    1.879646018  
    5.03  
    24  
    2.425  
    4.5  
    6
```

```
6.133333333
```

```
12
```

```
9
```

```
6
```

```
14
```

```
5.510204082
```

```
6
```

```
5.333333333
```

```
5.332075472
```

```
3.366666667
```

```
4.792
```

```
8
```

```
6.54109589
```

```
12.51428571
```

```
3.805
```

```
5.2
```

```
9.2
```

```
4.461504425
```

```
7.2
```

```
4.159060403
```

```
3.773953488
```

```
12.4080402
```

```
12.8
```

```
18.89230769
```

```
4.72244898
```

```
11.29133858
```

```
4.92278481
```

```
18.98666667
```

```
5.493333333
```

```
4.3
```

```
5.2
```

```
5.769333333
```

```
2.586075949
```

```
13.19298246
```

```
15.49565217
```

```
9.2
```

```
20.09090909
```

```
14.4
```

```
14
```

```
26
```

```
];
```

```
% 每日补货量
```

```
daily_replenishment = [
```

1
1
1
1
0.183914773
1
1
0.604096774
0.62025
0.404123288
0.512
0.541276596
0.401176471
1
0.44385
0.38945679
0.391830189
1
0.44959
0.409530303
1
0.467
0.520508333
0.381177778
0.324527778
1
0.392153846
1
0.432916279
2.025892663
1.919273725
0.219230769
1
0.377141732
1
0.2434
1
1
0.366301471
1
1
0.627984035
0.564913043

```

0.921961667
1.106272727
1.678980444
2.568347391
4.038208333
];

% 约束条件
min_display_quantity = 2.5; % 最小陈列量
min_selected_items = 27; % 选出的最小单品数量
max_selected_items = 33; % 选出的最大单品数量

% 粒子群算法参数
num_particles = 50; % 粒子数量
num_iterations = 100; % 迭代次数
w = 0.5; % 惯性权重
c1 = 1; % 个体认知因子
c2 = 1; % 社会认知因子

% 初始化粒子群
particle_position = rand(num_particles, 50); % 随机初始化粒子位置
particle_velocity = zeros(num_particles, 50); % 初始速度为0
particle_best_position = particle_position; % 记录粒子的最佳位置
global_best_position = particle_best_position(1, :); % 记录全局最佳位置

% 计算初始适应度
for i = 1:num_particles
    % 计算适应度，这里需要根据你的目标函数编写
    % 利润 = 每个单品的补货总量 * (每个单品的定价 - 每个单品的批发价格)
    % 这里需要将目标函数适应度计算代码替换成你的实际计算逻辑
    profit = sum(particle_position(i, :) .* (retail_prices - wholesale_prices));

    % 更新粒子最佳位置和全局最佳位置
    particle_best_position(i, :) = max(particle_best_position(i, :), profit);
    global_best_position = max(global_best_position, profit);
end

% 开始迭代
for iteration = 1:num_iterations
    for i = 1:num_particles
        % 更新粒子速度和位置
        r1 = rand(1, 50);
        r2 = rand(1, 50);
        particle_velocity(i, :) = w * particle_velocity(i, :) + ...

```

```

        c1 * r1 .* (particle_best_position(i, :) - particle_position(i, :))
+ ...
        c2 * r2 .* (global_best_position - particle_position(i, :));
    particle_position(i, :) = particle_position(i, :) +
particle_velocity(i, :);

    % 约束条件处理
    % 这里需要根据你的约束条件编写处理代码

    % 计算适应度
    profit = sum(particle_position(i, :) .* (retail_prices -
wholesale_prices));

    % 更新粒子最佳位置和全局最佳位置
    if profit > particle_best_position(i)
        particle_best_position(i, :) = profit;
    end
    if profit > global_best_position
        global_best_position = profit;
    end
end
end

% 输出结果
disp(['最终总利润最大值: ', num2str(global_best_position)]);
disp(['最优解: ', num2str(particle_position(1, :))]);

```

附录 4 Q3linear.m

介绍：通过一周的信息预测 7.1 批发价格

```

dates = 1:7; % 6 月 23 日到 6 月 30 日的日期
prices = [1
    2
    3
    4
    5
    6
    7];

% 创建日期矩阵，包括未来一周的日期
future_dates = 8:8; % 假设 6 月 31 日到 7 月 6 日的日期

% 使用线性回归模型进行预测

```

```

coefficients = polyfit(dates, prices, 1);
predicted_prices = polyval(coefficients, future_dates);

disp('预测的批发价格: ');
disp(predicted_prices);

% 可以将预测结果可视化
plot(dates, prices, 'o', future_dates, predicted_prices, '-');
xlabel('日期');
ylabel('批发价格');
legend('实际价格', '预测价格');
title('蔬菜批发价格预测');

```

附录 5 各个单品销量及价格的数据分析图及最终决策利润收入图

介绍：内包含每个单品各项数据，可用于预测回归 7 月 1 日的各项数据，最终得到利润收入图

单品	平均销量	平均利润	批发价格平 均值	销售价格平 均值
白玉菇(袋)	1	2.1975	3.5025	5.7
虫草花(份)	1	0.94625	2.66625	3.6125
海鲜菇(包)	1	0.795	1.953387097	2.748387097
金针菇(盒)	1	0.42628319	1.453362832	1.879646018
双孢菇(盒)	1	1.62742857	3.402571429	5.03
西峡花菇(1)	0.18391477	1.54488409	15.6	24
鲜木耳(份)	1	1.125	1.3	2.425
蟹味菇与白玉菇双 拼(盒)	1	1.25	3.25	4.5
青茄子(1)	0.60409677	1.15632774	4.082258065	6
圆茄子(2)	0.62025	1.35765083	3.8775	6.133333333
长线茄	0.40412329	2.02661329	6.984931507	12
紫茄子(1)	0.512	2.38592	4.34	9
紫茄子(2)	0.5412766	1.2285039	3.737021277	6
菠菜	0.40117647	1.75089118	9.635294118	14
菠菜(份)	1	1.37530612	4.134897959	5.510204082
菜心	0.44385	0.618278	4.606	6
红薯尖	0.38945679	0.83289037	3.190740741	5.333333333
木耳菜	0.39183019	0.85816745	3.222735849	5.332075472
木耳菜(份)	1	1.8	1.566666667	3.366666667
奶白菜	0.44959	1.0130036	2.5314	4.792
上海青	0.4095303	1.58481091	4.131969697	8
娃娃菜	1	1.78383562	4.757260274	6.54109589
外地茼蒿	0.467	1.99399286	8.241428571	12.51428571
苋菜	0.52050833	0.77539317	2.325583333	3.805
小青菜(1)	0.38117778	0.90346167	2.842222222	5.2
云南生菜	0.32452778	1.12296833	5.741666667	9.2
云南生菜(份)	1	0.88570796	3.57579646	4.461504425
云南油麦菜	0.39215385	1.10767846	4.34	7.2
云南油麦菜(份)	1	1.2895302	2.869530201	4.159060403
竹叶菜	0.43291628	0.62219488	2.330139535	3.773953488
西兰花	2.02589266	0.44172864	7.772964824	12.4080402
枝江青梗散花	1.91927373	0.49417647	8.945490196	12.8
红椒(2)	0.21923077	1.27562231	12.97323077	18.89230769
姜蒜小米椒组合装 (小份)	1	2.26367347	2.45877551	4.72244898
螺丝椒	0.37714173	1.44343173	7.481653543	11.29133858
螺丝椒(份)	1	1.59443038	3.32835443	4.92278481
七彩椒(2)	0.2434	1.661971	12.27466667	18.98666667
青红杭椒组合装(份)	1	2.28266667	3.210666667	5.493333333

7月1日最终各单品进货量与定价方案:

品类	单品	批发量 kg	销售价	利润 (元)
花菜类	西兰花	16.59545765	12.5665158	56.4211078
花菜类	枝江青梗散花	8.607581967	12.8	24.2863026
花叶类	菠菜(份)	3.10564663	14	9.92243137

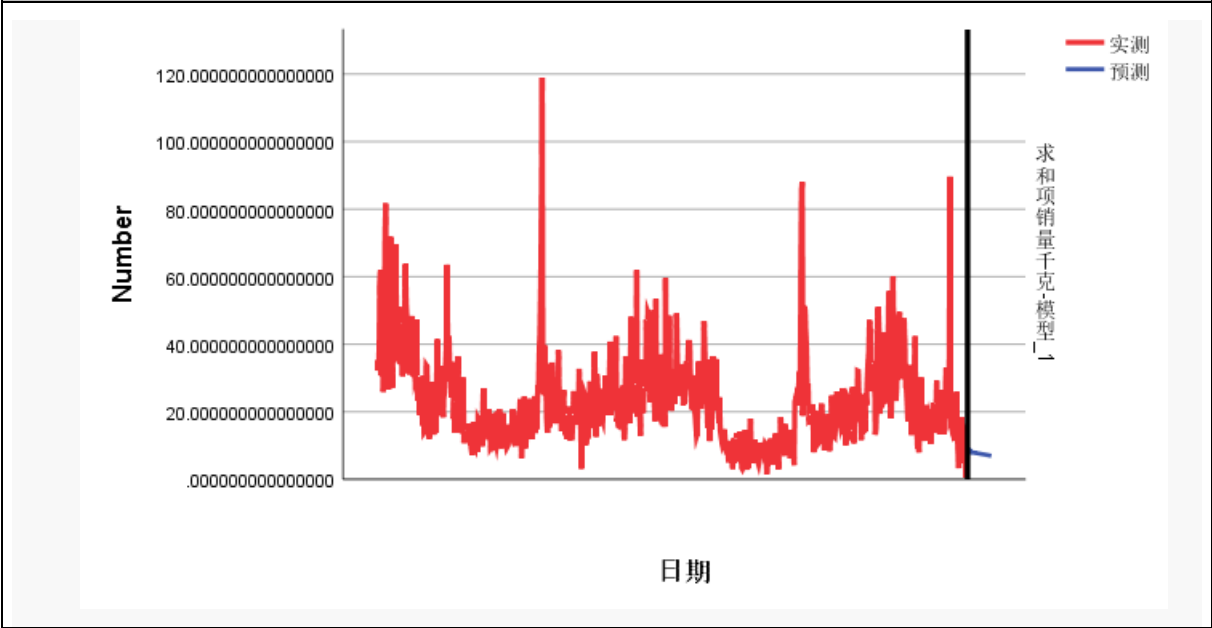
花叶类	红薯尖	7.25	5.48571429	11.6829815
花叶类	木耳菜	8.046618852	5.36470588	12.5474512
花叶类	奶白菜	11.50546448	4.88125	19.7930159
花叶类	上海青	5.417862022	8	15.3568073
花叶类	娃娃菜	12.29508197	6.53333333	16.1511111
花叶类	苋菜	12.06813525	3.82919708	13.1134683
花叶类	小青菜(1)	6.738558743	5.2	11.5384738
花叶类	云南生菜(份)	47.98497268	4.39217082	31.67125
花叶类	云南油麦菜(份)	28.85928962	4.16745562	27.9175
花叶类	竹叶菜	19.5	3.81307692	20.624283
辣椒类	姜蒜小米椒组合装(小份)	11.44125683	4.71492537	18.6225
辣椒类	螺丝椒	9.00034153	11.1468531	25.246819
辣椒类	螺丝椒(份)	17.41803279	4.69215686	19.1375
辣椒类	芜湖青椒(1)	19.46704235	5.2	26.8759058
辣椒类	小米椒(份)	26.81010929	5.77070064	70.9700001
辣椒类	小皱皮(份)	16.39344262	2.6125	13.54375
茄类	长线茄	47.70355191	12	173.238377
茄类	紫茄子(2)	16.15061475	6	26.5858758
食用菌	海鲜菇(包)	12.63661202	2.74054054	7.24125003
食用菌	金针菇(盒)	22.71174863	1.86165414	6.79625005
食用菌	双孢菇(盒)	14.85655738	5.07126437	19.40625

食用菌	西峡花菇(1)	6.452527322	24	39.6753
水生根茎类	洪湖藕带	5.649419399	20.5955056	10.733389
水生根茎类	净藕(1)	8.230874317	14.2654206	21.0221821

附录 6

9 月 12 日至 10 月 14 日茄类缺失值的预测图

介绍



单品	平均销量	平均利润	批发价格平 均值	销售价格平 均值
白玉菇(袋)	1	2.1975	3.5025	5.7
虫草花(份)	1	0.94625	2.66625	3.6125
海鲜菇(包)	1	0.795	1.953387097	2.748387097
金针菇(盒)	1	0.42628319	1.453362832	1.879646018
双孢菇(盒)	1	1.62742857	3.402571429	5.03
西峡花菇(1)	0.18391477	1.54488409	15.6	24
鲜木耳(份)	1	1.125	1.3	2.425
蟹味菇与白玉菇双 拼(盒)	1	1.25	3.25	4.5
青茄子(1)	0.60409677	1.15632774	4.082258065	6
圆茄子(2)	0.62025	1.35765083	3.8775	6.133333333
长线茄	0.40412329	2.02661329	6.984931507	12
紫茄子(1)	0.512	2.38592	4.34	9
紫茄子(2)	0.5412766	1.2285039	3.737021277	6
菠菜	0.40117647	1.75089118	9.635294118	14
菠菜(份)	1	1.37530612	4.134897959	5.510204082
菜心	0.44385	0.618278	4.606	6
红薯尖	0.38945679	0.83289037	3.190740741	5.333333333
木耳菜	0.39183019	0.85816745	3.222735849	5.332075472
木耳菜(份)	1	1.8	1.566666667	3.366666667
奶白菜	0.44959	1.0130036	2.5314	4.792
上海青	0.4095303	1.58481091	4.131969697	8
娃娃菜	1	1.78383562	4.757260274	6.54109589
外地茼蒿	0.467	1.99399286	8.241428571	12.51428571
苋菜	0.52050833	0.77539317	2.325583333	3.805
小青菜(1)	0.38117778	0.90346167	2.842222222	5.2
云南生菜	0.32452778	1.12296833	5.741666667	9.2
云南生菜(份)	1	0.88570796	3.57579646	4.461504425
云南油麦菜	0.39215385	1.10767846	4.34	7.2
云南油麦菜(份)	1	1.2895302	2.869530201	4.159060403
竹叶菜	0.43291628	0.62219488	2.330139535	3.773953488
西兰花	2.02589266	0.44172864	7.772964824	12.4080402
枝江青梗散花	1.91927373	0.49417647	8.945490196	12.8
红椒(2)	0.21923077	1.27562231	12.97323077	18.89230769
姜蒜小米椒组合装 (小份)	1	2.26367347	2.45877551	4.72244898
螺丝椒	0.37714173	1.44343173	7.481653543	11.29133858
螺丝椒(份)	1	1.59443038	3.32835443	4.92278481
七彩椒(2)	0.2434	1.661971	12.27466667	18.98666667
青红杭椒组合装(份)	1	2.28266667	3.210666667	5.493333333

7月1日最终各单品进货量与定价方案:

品类	单品	批发量 kg	销售价	利润 (元)
花菜类	西兰花	16.59545765	12.5665158	56.4211078
花菜类	枝江青梗散花	8.607581967	12.8	24.2863026
花叶类	菠菜(份)	3.10564663	14	9.92243137

花叶类	红薯尖	7.25	5.48571429	11.6829815
花叶类	木耳菜	8.046618852	5.36470588	12.5474512
花叶类	奶白菜	11.50546448	4.88125	19.7930159
花叶类	上海青	5.417862022	8	15.3568073
花叶类	娃娃菜	12.29508197	6.53333333	16.1511111
花叶类	苋菜	12.06813525	3.82919708	13.1134683
花叶类	小青菜(1)	6.738558743	5.2	11.5384738
花叶类	云南生菜(份)	47.98497268	4.39217082	31.67125
花叶类	云南油麦菜(份)	28.85928962	4.16745562	27.9175
花叶类	竹叶菜	19.5	3.81307692	20.624283
辣椒类	姜蒜小米椒组合装(小份)	11.44125683	4.71492537	18.6225
辣椒类	螺丝椒	9.00034153	11.1468531	25.246819
辣椒类	螺丝椒(份)	17.41803279	4.69215686	19.1375
辣椒类	芜湖青椒(1)	19.46704235	5.2	26.8759058
辣椒类	小米椒(份)	26.81010929	5.77070064	70.9700001
辣椒类	小皱皮(份)	16.39344262	2.6125	13.54375
茄类	长线茄	47.70355191	12	173.238377
茄类	紫茄子(2)	16.15061475	6	26.5858758
食用菌	海鲜菇(包)	12.63661202	2.74054054	7.24125003
食用菌	金针菇(盒)	22.71174863	1.86165414	6.79625005
食用菌	双孢菇(盒)	14.85655738	5.07126437	19.40625

食用菌	西峡花菇(1)	6.452527322	24	39.6753
水生根茎类	洪湖藕带	5.649419399	20.5955056	10.733389
水生根茎类	净藕(1)	8.230874317	14.2654206	21.0221821