Statistics and Application 统计学与应用, 2026, 15(4), 323-337
Published Online April 2026 in Hans. https://www.hanspub.org/journal/sa
https://doi.org/10.12677/sa.2026.154094

Hans汉斯

商超蔬菜降损增利策略研究

王修修

重庆建筑工程职业学院通识教育学院，重庆

收稿日期：2026年3月27日；录用日期：2026年4月17日；发布日期：2026年4月30日

摘　要

本文针对生鲜蔬菜零售行业激烈竞争，以某商超为研究对象，基于附件数据，运用Python完成数据清洗与预处理，构建数学模型以优化蔬菜动态定价与补货策略，提升商超收益。问题一计算2021年7月~2024年6月每日蔬菜利润及利润率，分析得出花叶类利润率最高且稳定，花菜类具季节性，2023年下半年花叶类与辣椒类受事件影响利润率两次上升。问题二分析销量规律，发现品类及单品均呈季节性，除茄类外，其余五类蔬菜销量显著正相关。问题三通过Spearman相关分析得出销售额与成本加成定价负相关，拟合函数并推算7天销售额与定价，进而计算利润，为经营决策提供依据。

关键词

蔬菜动态定价与补货策略，利润率，Spearman相关分析，拟合函数预测价格

Research on Strategies for Reducing Losses and Increasing Profits in Supermarket Vegetables

Xiuxiu Wang

School of General Education, Chongqing Jianzhu College, Chongqing

Received: March 27, 2026; accepted: April 17, 2026; published: April 30, 2026

Abstract

This article focuses on the intense competition in the fresh vegetable retail industry. Taking a certain supermarket as the research object, based on the data in the attached file, Python is used to complete data cleaning and preprocessing, and a mathematical model is constructed to optimize the dynamic pricing and replenishment strategies of vegetables, thereby increasing the supermarket's revenue. Question 1: Calculate the daily vegetable profits and profit rates from July 2021 to June 2024,

文章引用: 王修修. 商超蔬菜降损增利策略研究[J]. 统计学与应用, 2026, 15(4): 323-337.
DOI: 10.12677/sa.2026.154094

and analyze to find that the profit rates of leafy vegetables are the highest and stable, while cauliflower vegetables have a seasonal pattern. In the second half of 2023, the profit rates of leafy vegetables and chili vegetables both rose twice due to events. Question 2: Analyze the sales patterns and find that both categories and individual items show sea sonal patterns. Except for the eggplant category, the sales of the other five types of vegetables are significantly positively correlated. Question 3: Through Spearman correlation analysis, it is concluded that sales are negatively correlated with cost-plus pricing. A fitting function is proposed and the 7-day sales and pricing are calculated, thereby calculating the profit, providing a basis for business decisions.

Keywords

Dynamic Pricing and Replenishment Strategies of Vegetables, Profit Rate, Spearman Correlation Analysis, Fitting Function to Predict Prices

1. 引言

1.1. 问题背景

生鲜商超在提升盈利能力和服务质量方面，降低蔬菜损耗是一个关键挑战。根据农产品流通产业发展报告，我国果蔬、肉类和水产品的流通损耗率分别为25%、12%和15%，远高于欧美发达国家的5%。对于运输过程中受损或品相变差的商品，通常会采取打折销售的方式。由于蔬菜类商品的保鲜期较短，随着销售时间的延长，品相会逐渐变差，许多品种如果在当天未能售出，次日便无法再售。这种损耗直接影响到生鲜商超的营收和利润。因此，减少蔬菜损耗、提高盈利能力[1]，是生鲜商超亟待解决的重要问题。

1.2. 问题重述

附件1、2、3、4提供了某生鲜商超近三年蔬菜类商品的单品编码、类别等基本信息，以及进价、定价和销量等销售及进货数据。为了帮助该商超在未来蔬菜类商品的销售中做出明智决策，以实现更高的利润，我们需要结合这些实际数据和相关信息，构建一个数学模型，以分析并解决以下问题：

问题一：结合附件1，2，3，计算出该生鲜商超2021-7-1到2024-6-30日期里每天蔬菜的利润及利润率，并且从得出的数据中通过描述统计的方法和分析，以此判断该生鲜商超的各品类蔬菜的利润率分布规律。

问题二：依据附件1，2中给出的6种类蔬菜品类的商品信息以及销售流水明细，合并统计相关数据，分别分析生鲜商超的蔬菜各品类、蔬菜单品销售量的分布规律、相互关系，判断不同品类或不同单品之间是否存在一定的关联关系。

问题三：从附件1、2、3、4中计算出各蔬菜品类的成本加成定价，再分析其与各蔬菜品类销售总量之间的关系。通过预期未来一周(2023年7月1日~7日)的日补货量，以生鲜商超利润最大化为目标制定出各蔬菜品类定价策略。

2. 问题分析与模型假设

2.1. 问题一分析

针对问题一，第一小问，首先对分析附件2中有退货和打折出售的数据，会对计算利润及利润率产

生误差，所以判断为无效数据并且进行剔除处理，然后通过 python 的 pandas 库，对附件 2，3 进行数据的处理异常值的处理，然后与附件 1 统计为一个表，然后通过分类各类蔬菜的品类再进行对利润及利润率的计算。第二小问，可以在得到第一个小问，数据基础上，通过分析数据的描述统计值均值、最大值、最小值、中位数、标准差、偏度系数、峰度系数以及总销量的柱状图和日销量折线图来观测 2020 年 7 月 1 日~2023 年 6 月 30 日各品类蔬菜和单品蔬菜的销售情况及销售趋势，以此判断蔬菜各品类利润率的分布规律。

2.2. 问题二分析

针对问题二，第一小问，在问题一得到数据统计表再次进行分类汇总，以附件 1 中的蔬菜品类和单品种类两个定量加以划分求出每日的销售量，再次对得到的两个数据表进行描述性统计分析以及通过可视化分析绘制柱状图，饼图，折线图来分析蔬菜各类和各单品的分布规律。第二小问，通过 Spearman 相关系数，以得到的各蔬菜大类及大类中的单品蔬菜每日销售量为指标，进行蔬菜各品类销售量的相关性分析。然后卡方检验，以此来验证确定模型的准确性。

2.3. 问题三分析

针对问题三，第一小问，先根据成本加成定价公式计算得出各蔬菜品类三年来每日定价情况，然后引入 Pearson 相关系数判定各蔬菜品类的销售总量与成本加成定价的线性关系，最后建立线性回归模型，使用最小二乘法求出各蔬菜品类与其成本加成定价之间的线性回归方程。第二小问，针对不同品类蔬菜建立合适的时间序列预测模型，考虑到数据的时效性，本文首先以前半年日销售量为原始数据预测未来一周各蔬菜品类的日销售量，然后根据预测结果建立优化模型。最后引入神经网络推出求出最优的蔬菜品类的补货和定价策略。

2.4. 模型假设

1. 假设蔬菜的以前的销售数据能够代表未来的销售趋势，本文使用海鲜商超的以前销售数据来预测未来的销售。

2. 假设销量不会受到人为影响波动，如竞争对手的恶意降价。

3. 假设短期内商超的进货与售价不会对顾客忠诚度造成影响。

3. 模型建立与求解

3.1. 问题一模型建立与求解

3.1.1. 数据预处理

除在附件 2 中的数据存在退货和打折出售的情况，退货数值的销售的单价为负数，如果带入计算则会对计算利润及利润率造成影响，不能够准确反映利润及利润率，所以利用 python 先将附件 2 中的退货和打折数据全部移除掉，把附件 2 和附件 3 进行异常值处理，然后再合并附件 1，2，3 到统一表中以便于分类数据处理(样表)如表 1 所示。

Table 1. Consolidated data table
表 1. 合并后的数据表

销售日期	单品编码	销量(千克)	…	分类编码	分类名称
2021-07-01	102900005117056	0.432		1011010504	辣椒类

续表

2021-07-01	102900005115960	0.925	1011010101	花叶类
2021-07-01	102900005117056	0.446	1011010504	辣椒类
2021-07-01	102900005115823	0.459	1011010101	花叶类
⋮				
2024-06-30	102900011016701	0.275	1011010504	辣椒类
2024-06-30	102900011022764	0.875	1011010501	茄类

3.1.2. 计算利润及利润率

对于第一问的计算利润及利润率，通过查阅的文献得知利润率的计算公式为[2]

$$S_{\text{利润率}} = S_{\text{利润}} \div S_{\text{成本}} \times 100\% \tag{1}$$

所以用 Python 中的 pandas 按日期分类计算求出合并数据每天的成本以及利润，最后根据公式(1)求出从 2020 年 7 月 1 日~2023 年 6 月 30 日利润率。(Python 代码见附录)

3.1.3. 通过数据分析分布规律

对于第二问，要求分析分析蔬菜各品类利润率的分布规律，本文通过对数据进行描述性统计分析，以及可视化识图的分析来判断蔬菜各品类利润率的分布规律。

(1) 描述性统计分析

描述统计是通过图表或数学方法，对数据资料进行整理、分析，并对数据的分布状态、数字特征和随机变量之间关系进行估计和描述的方法。描述统计分为集中趋势分析和离中趋势分析和相关分析三大部分[3]。选用中心趋势，离散程度，分布形状 3 种度量来分析分布规律，包括均值、最大值、最小值、中位数、标准差、偏度系数、峰度系数[4]来分析规律。利用 Python 求解描述统计值如表 2 所示。

Table 2. The daily profit margin of the six categories of vegetables describes the statistical values
表 2. 蔬菜六大类日利润率描述统计值

指标	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
均值	0.665	1.148	0.769	0.819	0.974	0.862
中位数	0.644	1.047	0.706	0.771	0.830	0.840
标准差	0.235	0.468	0.337	0.267	0.861	0.205
最小值	0.124	0.602	0.140	−0.042	−0.035	0.269
最大值	6.188	9.805	3.080	2.622	20.770	2.028
偏度	12.113	8.237	3.417	1.822	13.422	1.403
峰度	280.337	127.065	16.364	7.132	274.223	4.863

通过计算结果可知，蔬菜的六大品类中，花叶类均值最较高，标准差适中，说明利润率是 6 类中情况较稳定并且利润较高的。六大品类中所有的偏度值都大于 0，说明数据分布相对于均值向右倾斜，即存在一些较高销售量的极端值。辣椒类的偏度值最小，表明其利润率分布相对于其他品类更加靠近均值。辣椒类和水生根茎类峰度值和偏度最高，说明其利润率分布尖峰程度在 6 类中最窄高并且右偏严重，偏离正态分布最远。

(2) 数据可视化识图分析

数据可视化主要旨在借助于图形化手段，清晰有效地传达与沟通信息。在通过描述性统计分析后进行可视化分析能更深入了解各类蔬菜的分布规律。

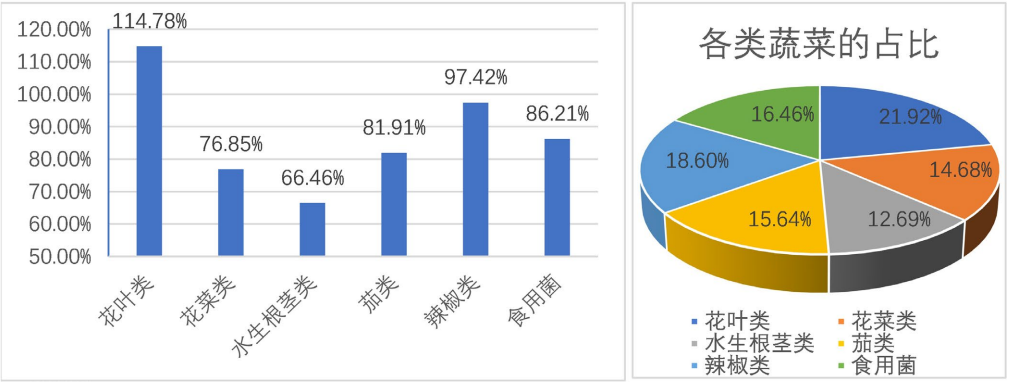

Figure 1. Distribution chart of total profit margins of six categories of vegetables
图 1. 蔬菜六大类总利润率分布图

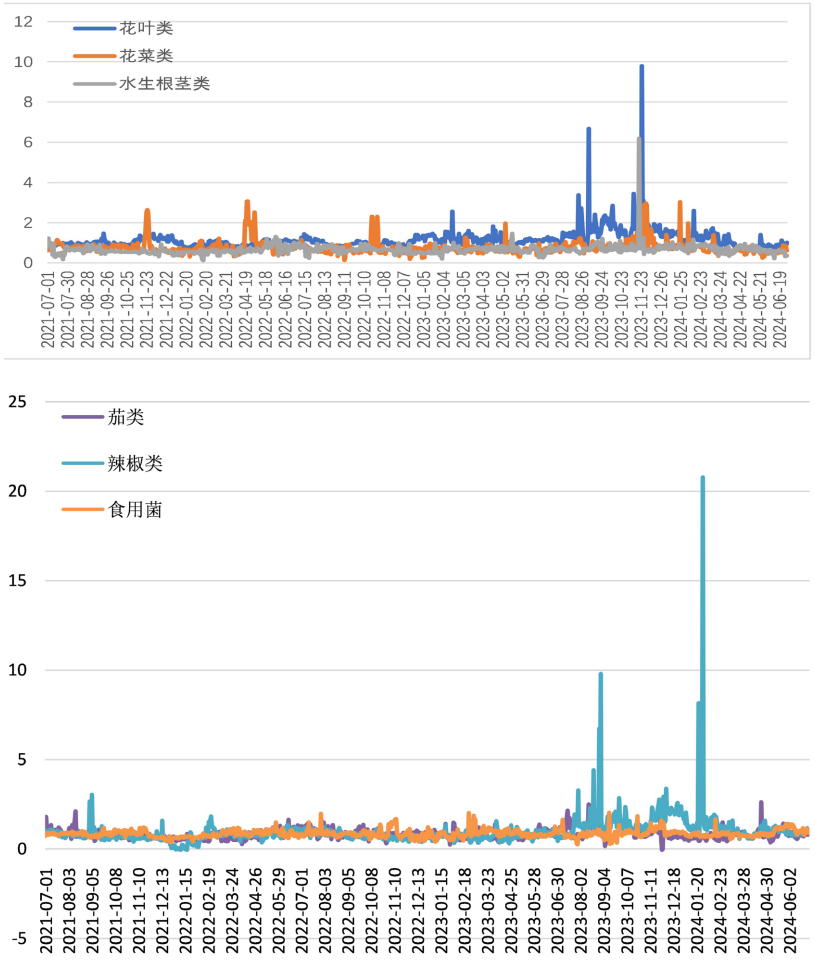

Figure 2. Line chart of daily profit margins of six categories of vegetables
图 2. 蔬菜六大类日利润率折线图

从图1中可以看出6大蔬菜类各自的利润率以及总的占比，其中花叶类蔬菜的利润率比高达114.78%，并且占比也到了20%以上，是利润率最大的蔬菜总类，而水升根茎类则为利润率最低。

从图2中可以看出6大蔬菜的每日利润率，花菜类呈现了季节性的顶峰，说明因为季节性因素导致。在其中花叶类与辣椒类在2023下半年里面出现了两次凸增，可能是因为特殊事件所导致。其余数据呈现较平稳的分布规律。

3.2. 问题二模型建立与求解

3.2.1. 通过数据分析分布规律

(1) 描述性统计分析

在问题一中，已经求解出了各类蔬菜的利润率，所以再次用Python进行数据描述性统计分析得到各类蔬菜销售量如表3所示。

Table 3. The daily sales volume of six categories of vegetables describes the statistical values
表3. 蔬菜六大类日销售量描述统计值

指标	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
均值	37.128	186.107	39.806	22.674	82.262	70.106
中位数	27.726	174.812	35.247	19.944	71.781	54.793
标准差	33.872	88.686	24.930	13.992	51.892	51.874
最小值	0.531	34.119	0.689	0.275	6.612	3.283
最大值	323.504	1322.684	202.913	129.632	647.717	557.164
偏度	2.617	2.990	1.508	1.776	3.951	3.224
峰度	13.047	28.064	4.347	6.380	28.106	18.731

由表3得知，蔬菜的6大类中花叶类的均值远高于其他种类，但是标准差和峰度相对其他类也大，所以花叶类销售量不稳定。

由于单品蔬菜种类颇多，本文表4选取各类蔬菜中总销售量前6的单品蔬菜(详表见附录)。

Table 4. The daily profit margin of the six categories of vegetables describes the statistical values
表4. 蔬菜六大类日利润率描述统计值

指标	枝江青梗散花	金针菇(盒)	紫茄子(2)	上海青	芜湖青椒	莲蓬(个)
均值	0.513	1.091	0.540	0.498	1.091	0.405
中位数	0.486	1.090	0.507	0.464	23.14	0.000
标准差	0.152	0.054	0.255	0.240	0.023	0.174
最小值	0.134	1.090	0.117	0.044	0.000	0.000
最大值	1.440	4.360	13.625	9.808	249.97	80.00
偏度	1.163	48.079	6.816	4.767	2.88	5.54
峰度	2.347	2523.866	272.019	140.439	20.57	37.63

由表4可得金针菇和芜湖青椒的均值较高并且标准差也低，所以该两类销售量稳定，并且所有的蔬

菜偏度都大于 0，呈现正态分布。

(2) 数据可视化识图分析

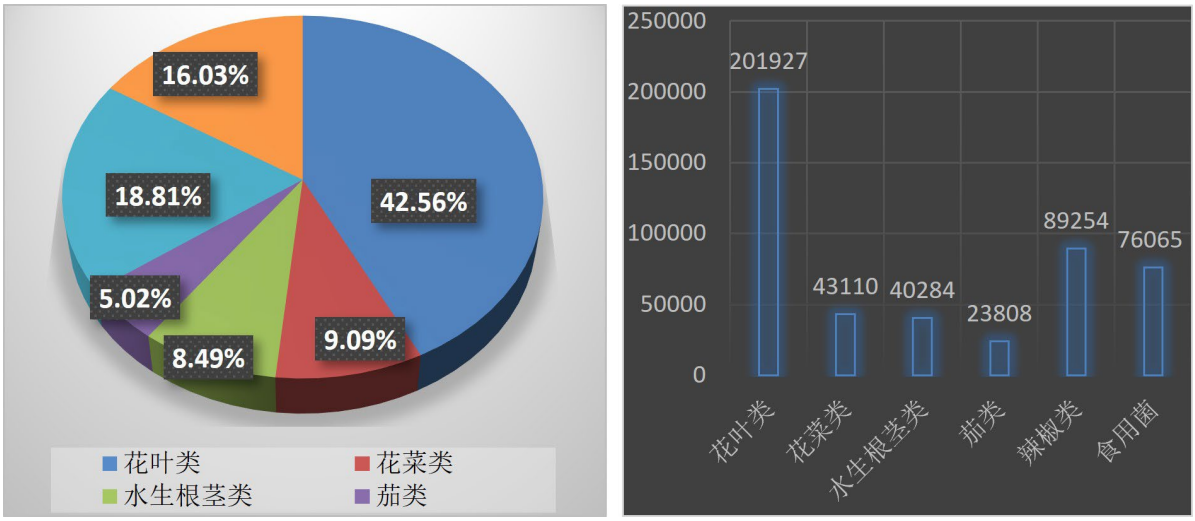

Figure 3. Distribution map of the total sales volume of six categories of vegetables
图 3. 蔬菜六大类总销售量分布图

从图 3 中可以直观地看出花叶类的销售量占比高达 42%，为 6 类中最高的，其次是辣椒类和食用菌类，最少的是茄类。

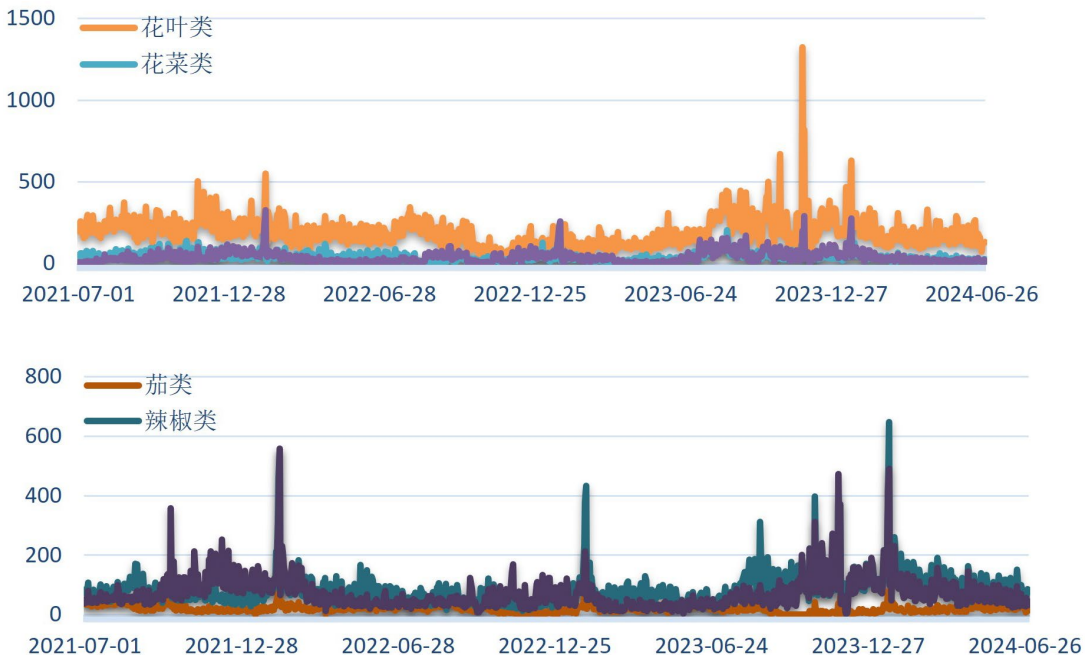

Figure 4. Line chart of daily sales volume of six categories of vegetables
图 4. 蔬菜六大类日销售量折线图

由图 4 可知，各类蔬菜在每年 12 月时会出现销量大幅增加，呈现出季节性分布的销售量规律。在这

其中食用菌类与辣椒类在 12 月份之外也有呈现销量大幅增加的情况，说明这两类蔬菜销量受特殊事件影响巨大。

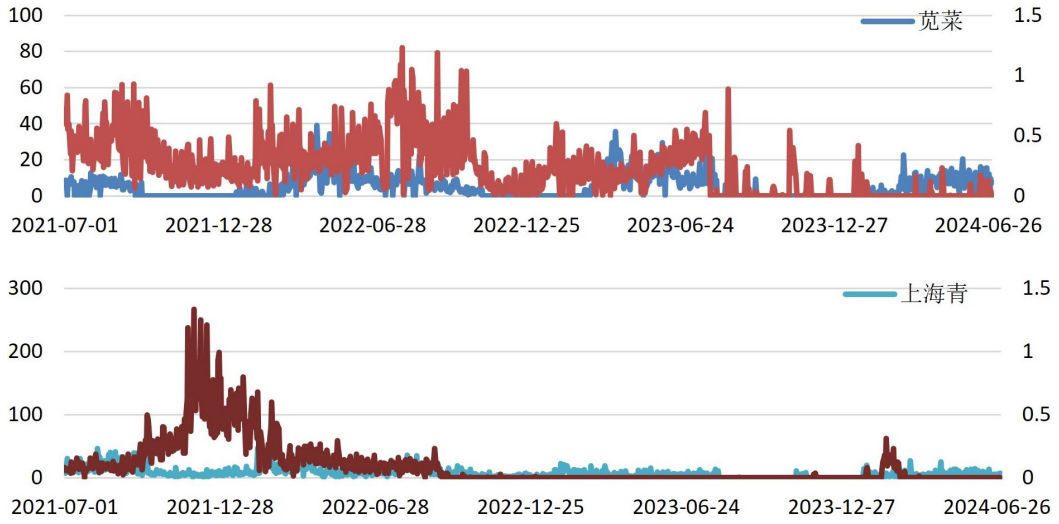

Figure 5. Distribution of single-product vegetables
图 5. 单品蔬菜的分布情况

由于篇幅限制，不能将所有单品数据依次进行对比，所以挑选苋菜与云南生菜，上海青与大白菜进行分布分析。从图 5 中可以明显看到每个单品都有对应的季节增长销售，呈现季节性分布。

3.2.2. 分别建立模型判断关联关系

(1) Spearman 相关系数的运用

在统计学中，以查尔斯·爱德华·斯皮尔曼命名的斯皮尔曼等级相关系数，即 spearman 相关系数[5]。经常用希腊字母 ρ 表示。它是衡量两个变量的依赖性的非参数指标。它利用单调方程评价两个统计变量的相关性。Spearman 相关性的基本思想是：分别对两个变量 X、Y 做等级变换(rank transformation)，用等级 RX 和 RY 表示；然后按 Pearson 相关性分析的方法计算 RX 和 RY 的相关性。如果数据中没有重复值，并且当两个变量完全单调相关时，斯皮尔曼相关系数则为+1 或−1 [4]。

斯皮尔曼相关系数被定义成等级变量之间的皮尔逊相关系数。对于样本容量为 n 的样本，n 个原始数据被转换成等级数据，相关系数 ρ 为

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{2}$$

首先对 X，Y 集合同时降序或升序排列，得到两个元素排序集合 x、y，其中元素 x_i，y_i 分别为 X_i，Y_i。在各自集合中的排序。设定 d 集合为 X 与 Y 集合中相同位元素排序之差，d 集合各元素计算式如下：

$$d_i = x_i - y_i \tag{3}$$

实际应用中，变量间的连结是无关紧要的，于是可以通过简单的步骤计算 ρ 被观测的两个变量的等级的差值，则 ρ 为

$$\rho = 1 - \frac{6\sum d_i^2}{n\left(n^2 - 1\right)} \tag{4}$$

利用 SPSS 求解蔬菜六大品类的 Spearman 相关系数如下表所示：

Table 5. Spearman correlation coefficients of six categories of vegetables
表 5. 蔬菜六大品类的 Spearman 相关系数

			花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌
斯皮尔曼 Rho	花叶类	相关系数	1.000	0.650**	0.495**	0.331**	0.592**	0.587**
	花菜类	相关系数	0.650**	1.000	0.477**	0.208**	0.501**	0.504**
	水生根茎类	相关系数	0.495**	0.477**	1.000	−.108**	0.439**	0.577**
	茄类	相关系数	0.331**	0.208**	−0.108**	1.000	0.210**	−0.043
	辣椒类	相关系数	0.592**	0.501**	0.439**	0.210**	1.000	0.522**
	食用菌	相关系数	0.587**	0.504**	0.577**	−0.043	0.522**	1.000

注：**0.在 0.01 级别(双尾)，相关性显著。

表 5 可得，花菜类与其它类的相关系数较高，呈正相关性，其中花叶类与花菜类的相关系数 ρ 为 0.65，表示他们之间存在较强的正向相关关系。食用菌与花菜类，花叶类，辣椒类和水生根茎类呈正相关关系，且相关性较高。但与茄类的 Spearman 相关系数为−0.043，相关性很差。花叶类与其它品类的相关系数较高，表明它们之间存在较强的正相关关系。辣椒类于其它品类具有较强的正相关关系。茄类和其它蔬菜品类之间的相关系数较低，说明茄类与其他品类之间的销售量关系较弱。

3.3. 问题三模型建立与求解

3.3.1. 成本加成定价

成本加成定价法是按产品单位成本加上一定比例的利润制定产品价格的方法。也就是在产品成本上增加一部分盈利的方法。大多数企业是按成本利润率来确定所加利润的大小的。即：价格 = 单位成本 + 单位成本 × 成本利润率 = 单位成本(1 + 成本利润率)完全成本加成定价法是企业较常用的定价方法[5]。

$$S_{价格} = S_{单位成本} + S_{单位成本} * S_{损耗率} + S_{成本利润率} \tag{5}$$

3.3.2. Spearman 相关系数判断线性关系

通过 SPSS 对销售量和成本加成定价的 Spearman 相关性分析，得出表 6。

Table 6. Spearman correlation coefficient table for sales volume and markup cost pricing
表 6. 销售量与加成成本定价[5]的 Spearman 相关系数表

相关性				
			单品销售总量	定价
斯皮尔曼 Rho	单品销售总量	相关系数	1.000	−0.266**
		显著性(双尾)	.	0.000
	定价	相关系数	−0.266**	1.000
		显著性(双尾)	0.000	.

注：**.在 0.01 级别(双尾)，相关性显著。

3.3.3. 制定销售计划

通过公式(5)计算各时间所对应的成本加成定价，将销量与其对应的成本加成定价通过 Matlab 拟合工具箱进行拟合函数的计算，得到对应的拟合函数。

$$\begin{cases} f(x) = a_0 + a_1 * \cos(x * w) + b_1 * \sin(x * w) \\ a_0 = 9.993(7.654, 1233) \\ a_1 = -2.661(-5.055, -0.26666) \\ b_1 = 1.756(-1.45, 4.961) \\ w = 3.541 * 10^{-5}\left(3.109 * 10^{-5}, 3.975 * 10^{-5}\right) \\ \text{置信区间}(95\%\text{的范围}) \end{cases} \tag{6}$$

推断未来一周的日需求量，再通过公式 6 得到推断的预测值，见表 7。

Table 7. Estimated demand value
表 7. 需求量推测值

花菜类	花叶类	辣椒类	茄类	水生根茎类	食用菌
42.852	230.746	92.137	12.020	52.745	105.558
43.495	223.002	85.924	15.152	48.173	97.900
43.848	217.617	83.410	17.446	45.538	92.162
44.042	213.873	82.393	19.127	44.018	87.863
44.148	211.270	81.982	20.357	43.143	84.642
44.207	209.459	81.815	21.259	42.638	82.228
44.239	208.201	81.748	21.919	42.347	80.420

表 7 带入公式(6)得到定价推测值以及利润，见表 8。

Table 8. Speculative value of pricing
表 8. 定价推测值

花菜类	花叶类	辣椒类	茄类	水生根茎类	食用菌
14.952	7.597	12.226	10.612	18.049	17.097
13.056	11.068	13.189	12.663	14.230	17.450
12.300	10.625	11.767	12.586	12.889	17.657
14.512	11.453	13.926	14.876	14.141	17.636
11.648	8.220	15.395	12.872	14.817	18.083
11.642	11.771	15.958	12.815	11.765	14.197
14.917	11.379	15.359	13.493	11.536	15.896
3528.762	3054.971	1442.097	635.987	449.753	273.321

4. 模型分析与推广

4.1. 模型优点

数据处理严谨：采用 Python 工具对多附件数据进行异常值、无效数据剔除及合并处理，结合描述性统计与可视化分析，确保数据基础可靠，为后续建模提供高质量支撑。

分析维度全面：从利润率分布、销售量规律、品类相关性到定价与销量关系，覆盖蔬菜经营核心环节，且聚焦六大蔬菜品类及重点单品，结论针对性强。

方法科学适配：灵活运用 Spearman 相关分析、时间序列预测、神经网络等方法，贴合生鲜蔬菜季节性、短保鲜期的特性，定价与补货决策更具实操性。

目标导向清晰：以利润最大化为核心目标，通过拟合函数、优化模型实现定量分析，输出的决策方案可直接服务于商超经营提质。

4.2. 模型缺点

假设条件局限：假设历史销售数据可代表未来趋势、不受人为因素干扰等，未考虑突发市场波动、消费偏好变化等不确定性因素，现实适用性受一定限制。

数据依赖度高：模型效果高度依赖附件数据的完整性与准确性，若实际经营中数据采集不全面(如未纳入天气、促销活动等影响因素)，可能导致决策偏差。

单品覆盖不足：仅选取各类蔬菜中总销量前 6 的单品进行分析，未涵盖全部单品特性，对小众单品的指导作用有限。

动态调整不足：虽提及动态定价，但未充分考虑蔬菜新鲜度随时间衰减的实时影响，缺乏分时段精准调价机制。

4.3. 模型推广

适用场景拓展：可推广至水果、鲜肉等其他短保鲜期生鲜品类，通过调整利润率计算、销量预测的参数阈值，适配不同品类的流通特性。

多业态适配：除连锁商超外，可应用于社区生鲜店、线上生鲜平台，针对不同业态的进货规模、消费群体特征，微调时间序列预测的周期与定价策略的弹性系数。

区域适配优化：结合不同地区的气候差异、饮食偏好，调整季节性分析权重，例如南方地区可强化水生根茎类蔬菜的季节波动系数，北方地区侧重花菜类等耐寒品类的规律适配。

技术融合升级：与商超 ERP 系统、库存管理系统对接，实时抓取销售、库存数据，实现模型自动迭代与决策动态更新；融入物联网技术监测蔬菜新鲜度，优化分时段定价逻辑。

行业标准参考：模型的数据分析框架与决策思路可为生鲜零售行业提供标准化分析模板，助力中小商超降低数据分析门槛，提升行业整体降损增利水平。

参考文献

[1] 邓玲丽, 陈绍慧, 李江阔, 等. 我国蔬菜产后损失现状、原因及对策[J]. 保鲜与加工, 2018, 18(6): 1-7.

[2] 斌权. 销售利润率计算公式应规范统一[J]. 工业会计, 2000(9): 23-24.

[3] 高景德, 王祥珩. 交流电机的多回路理论[J]. 清华大学学报, 1987, 27(1): 1-8.

[4] 茆诗松, 程依明, 濮晓龙. 概率论与数理统计教程(第 3 版) [M]. 北京: 高等教育出版社, 2019.

[5] 韩俊华, 干胜道. 成本加成定价法评介[J]. 财会月刊(会计版), 2012(8): 74-75.

附　录

(1) Python 代码：

```
数据预处理
import pandas as pd
import os
import numpy as np
from scipy.interpolate import CubicSpline
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from openpyxl import Workbook
from openpyxl.styles import NamedStyle, Font, PatternFill, Border, Side
from openpyxl.utils.dataframe import dataframe_to_rows
from sklearn.impute import KNNImputer
#获取脚本所在目录
script_dir = os.path.dirname(os.path.abspath(__file__))
#构建文件路径
file_path1 = os.path.join(script_dir, ‘附件 1.xlsx’)
file_path2 = os.path.join(script_dir, ‘附件 2.xlsx’)
file_path3 = os.path.join(script_dir, ‘附件 3.xlsx’)
file_path4 = os.path.join(script_dir, ‘附件 4.xlsx’)
#读取 Excel 文件
data1 = pd.read_excel(file_path1)
data2 = pd.read_excel (file_path2, usecols = [‘销售日期’, ‘单品编码’, ‘销量(千克)’, “销售单价(元/千克)”, ‘销售类型’, ‘是否打折销售’])
data3 = pd.read_excel (file_path3, usecols = [‘日期’, ‘单品编码’, ‘批发价格(元/千克)’])
data4 = pd.read_excel (file_path4)
print (“Excel 文件读取完成”)
#过滤掉退货数据
data2 = data2[data2['销售类型'] != '退货']
data2 = data2[data2['是否打折销售'] != '是']
data2 = pd.DataFrame(data2)
data3 = pd.DataFrame(data3)
print("退货数据过滤完成")
#  将销售日期转换为标准日期格式
data2['销售日期'] = pd.to_datetime(data2['销售日期']).dt.strftime('%Y-%m-%d')
data3.rename(columns={'日期': '销售日期'}, inplace=True)
data3['销售日期'] = pd.to_datetime(data3['销售日期']).dt.strftime('%Y-%m-%d')
```

```
print("销售日期转换完成")
# 处理 data2 中的异常值
def handle_outliers(x):
Q1 = x.quantile(0.25)
Q3 = x.quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
x[(x < lower_bound) | (x > upper_bound)] = np.nan
return x
data2['销售单价(元/千克)'] = data2.groupby('单品编码')['销售单价(元/千克)'].transform(handle_outliers)
print("data2 异常值处理完成")
# 处理 data3 中的异常值
data3['批发价格(元/千克)'] = data3.groupby('单品编码')['批发价格(元/千克)'].transform(handle_outliers)
print("data3 异常值处理完成")
# 计算销售额
data2['销售额'] = data2['销量(千克)'] * data2['销售单价(元/千克)']
print("销售额计算完成")
# 合并数据
alldata = pd.merge(data2, data3, on=['销售日期', '单品编码'], how='left')
alldata = pd.merge(alldata, data1, on=['单品编码'], how='left')
print("数据合并完成")
# 计算成本、利润和利润率
alldata['成本'] = alldata['批发价格(元/千克)'] * alldata['销量(千克)']
alldata['利润'] = alldata['销售额'] - alldata['成本']
alldata['利润率'] = alldata['利润'] / alldata['成本']
print("成本、利润和利润率计算完成")
# 按销售日期汇总利润和利润率
res1_1 = alldata.groupby('销售日期').agg({'利润': 'sum', '利润率': 'mean'}).reset_index()
print("按销售日期汇总利润和利润率完成")
# 按销售日期和分类编码汇总利润率
res1_2 = alldata.groupby(['销售日期', '分类编码']).agg({'利润率': 'mean'}).reset_index()
res1_2 = pd.merge(res1_2, data1[['分类编码', '分类名称']].drop_duplicates(), on='分类编码', how='left')
print("按销售日期和分类编码汇总利润率完成")
res2_1 = alldata.groupby(['销售日期', '分类编码']).agg({'销量(千克)': 'sum'}).reset_index()
res2_1 = pd.merge(res2_1, data1[['分类编码', '分类名称']].drop_duplicates(), on='分类编码', how='left')
print("按销售日期和分类编码汇总销量完成")
res2_2 = alldata.groupby(['销售日期', '单品编码']).agg({'销量(千克)': 'sum'}).reset_index()
res2_2 = pd.merge(res2_2, data1[['单品名称', '单品编码']].drop_duplicates(), on='单品编码', how='left')
print("按销售日期和单品编码汇总销量完成")
```

```
res3=alldata.groupby(['销售日期', '单品名称']).agg({'销售额': 'sum'}).reset_index()
res3=pd.merge(res3,alldata[['销售日期','单品名称','利润','批发价格(元/千克)','分类编码']].drop_duplicates(),on=['销售日期','单品名称'],how='left')
res3=pd.merge(res3, data4[['小分类编码','平均损耗率(%)_小分类编码_不同值']].drop_duplicates(), left_on='分类编码',right_on='小分类编码', how='left')
res3.rename(columns={'平均损耗率(%)_小分类编码_不同值':'损耗率','小分类编码':'分类编码'},inplace=True)
res3=res.groupby(['销售日期','分类名称']).agg({'利润':'sum','批发价格(元/千克)':'mean','损耗率':'mean'}).reset_index()
res3.to_excel('每日成本加 q 成 w 定价.xlsx', index=False)
print("每日分类利润保存完成")
# 创建一个空的 DataFrame，用于存储最终的网格图数据
unique_dates = res1_2['销售日期'].unique()
unique_categories = res1_2['分类名称'].unique()
res1= pd.DataFrame(index=unique_dates, columns=['日期'] + list(unique_categories))
res1['日期'] = res1.index
print("网格图数据初始化完成")
# 遍历每个日期和分类，将利润率填充到新的 DataFrame 中
for date in unique_dates:
for category in unique_categories:
profit_rate = res1_2[(res1_2['销售日期'] == date) & (res1_2['分类名称'] == category)]['利润率'].mean()
res1.loc[date, category] = profit_rate
print("利润率填充完成")
# 将结果保存到新的 Excel 文件中
res1.to_excel('每类每日利润率.xlsx', index=False)
print("每类每日利润率保存完成")
# 创建一个空的 DataFrame，用于存储最终的网格图数据
unique_dates = res2_1['销售日期'].unique()
unique_categories = res2_1['分类名称'].unique()
grid_data = pd.DataFrame(index=unique_dates, columns=['日期'] + list(unique_categories))
grid_data['日期'] = grid_data.index
print("网格图数据初始化完成")
# 遍历每个日期和分类，将销售额填充到新的 DataFrame 中
for date in unique_dates:
for category in unique_categories:
sales = res2_1[(res2_1['销售日期'] == date) & (res2_1['分类名称'] == category)]['销量(千克)'].sum()
grid_data.loc[date, category] = sales
print("销量填充完成")
# 将结果保存到新的 Excel 文件中
grid_data.to_excel('每类每日销量.xlsx', index=False)
print("每类每日销量保存完成")
```

```
# 建一个空的 DataFrame，用于存储最终的网格图数据
unique_dates = res2_2['销售日期'].unique()
unique_categories = res2_2['单品名称'].unique()
grid_data1 = pd.DataFrame(index=unique_dates, columns=['日期'] + list(unique_categories))
grid_data1['日期'] = grid_data1.index
print("网格图数据初始化完成")
# 遍历每个日期和分类，将销售额填充到新的 DataFrame 中
for date in unique_dates:
for category in unique_categories:
sales = res2_2[(res2_2['销售日期'] == date) & (res2_2['单品名称'] == category)]['销量(千克)'].sum()
grid_data1.loc[date, category] = sales
print("销量填充完成")
# 将结果保存到新的 Excel 文件中
grid_data1.to_excel('每品种每日销量.xlsx', index=False)
print("每品种每日销量保存完成")
#计算蔬菜各品类利润率的描述统计量
def calculate_descriptive_stats(data, category_col, profit_rate_col):
stats = data.groupby(category_col)[profit_rate_col].agg([
'mean', 'median', 'std', 'min', 'max',
lambda x: x.skew(),
lambda x: x.kurtosis()
])
stats.columns = ['均值', '中位数', '标准差', '最小值', '最大值', '偏度', '峰度']
return stats.T
res1_stats = calculate_descriptive_stats(res1_2, '分类名称', '利润率')
res1_stats.to_excel('蔬菜各品类利润率描述统计量.xlsx')
print("蔬菜各品类利润率描述统计量保存完成")
res2_1stats = calculate_descriptive_stats(res2_1, '分类名称', '销量(千克)')
res2_1stats.to_excel('蔬菜各品类销量描述统计量.xlsx')
print("蔬菜各品类销量描述统计量保存完成")
res2_2stats = calculate_descriptive_stats(alldata, '单品名称', '销量(千克)')
res2_2stats.to_excel('蔬菜各品种销量描述统计量.xlsx')
print("蔬菜各品种销量描述统计量保存完成")
```