基于 FB-Prophet 算法和非线性规划的蔬菜定价与补货决策

摘　要

针对问题一，本文首先对附件中的数据进行预处理。首先，将附件一和附件二进行整合，对各单品和各蔬菜类商品的分类的销量按不同月份进行加总，得到从各单品和各蔬菜类商品的分类的月度销量。接着，对各蔬菜类商品分类月度销量数据进行**归一化处理**，使得之后的相关性分析更加精确，降低不同类蔬菜商品之间重量差距带来的影响。对于第一小问，题目要求得出蔬菜商品各品类和单品销售量的分布规律，本文对蔬菜商品各品类的销量进行可视化，利用**折线图**直观地展现出蔬菜各品类的销量在不同月份和一天内的不同时间点的分布规律。利用**箱线图**描述各品类蔬菜的中位数、异常值和分布区间，分析出不同种类蔬菜销量的分布情况和规律。利用 SPSS 求解六类蔬菜类商品的**描述性统计**量，引入极差、平均值、标准差、偏度、峰度来描述统计量。本文使用 R 语言对不同蔬菜类商品的分类进行**系统聚类**，使用树状图和聚类谱系图表示系统聚类结果，分析蔬菜单品销售量的分布规律。对于第二小文，题目要求得出蔬菜各品类和单品销售量的相互关系。本文通过绘制蔬菜商品各品类之间的散点图矩阵和相关系数热力图，结合各蔬菜分类的**斯皮尔曼相关性系数**，分析出各蔬菜之间的相关性和相关强度。

针对问题二，根据过去三个月即 2023 年 4 月到 6 月一个季度中各蔬菜品类的销售量、销售价格和批发价格，进行 OLS 回归分析销售总量和成本加成定价之间的关系，接着对于过去三个月的销售量、销售价格和批发价格数据，使用 **FB-Prophet 时间序列预测方法**，预测出 7 月 1 日至 7 月 7 日各蔬菜品类的批发价和销售价格，同时利用已知的过去三个月的销售价格数据和销售量数据，为每个品类拟合销售价格和销量的数据，最后加以题目中所要求的限制条件，得到使得生鲜商超**收益最大化**的**补货策略**和**定价策略**。

针对问题三，首先根据 6 月 24 日至 6 月 30 日可以销售的蔬菜单品对所有蔬菜单品进行**初筛**，接着根据过去三个月即 2023 年 4 月到 6 月一个季度中各蔬菜单品的销售量、销售价格和批发价格，使用 **FB-Prophet 时间序列预测方法**，预测出 7 月 1 日各蔬菜单品的批发价和销售价格，同时利用已知的销量和售价数据拟合售价和销量之间的函数关系式，同时考虑到题目中要求蔬菜单品种类**控制在 27-33 个**，且各单品**订购量满足最小陈列量 2.5 千克**的限制条件，使用优化算法，得到使得生鲜商超收益最大化的以单品为单位的**补货总量**和**定价策略**。

针对问题四，题目要求寻找影响蔬菜商品补货和定价决策的其他相关数据，并给出支撑理由。影响商超的补货决策和定价点的因素有很多，本文从供应商、商家、消费者的角度进行了简要分析，并利用 Vensim 绘制出了因果回路图。从上述影响因素中，本文主要选取库存量和天气状况分析该数据的获取对完善商家补货和定价决策的影响，通过绘制**因果回路图**分析库存和天气状况对蔬菜补货量和定价的影响路径。

关键词：归一化 斯皮尔曼相关系数 FB-Prophet 预测 因果回路图

一、 问题重述

1.1 问题背景

在生鲜商超中，一般而言，蔬菜类商品的保鲜期较短。随着销售时间的增加，品质会逐渐恶化，如果大部分蔬菜品种在当日未售出，隔日就无法再次售卖。所以生鲜商超会依据各蔬菜单品的历史销售和需求情况进行每天的补货。

由于蔬菜种类多、场地不同，但是进货时间通常在凌晨 3：00-4：00，因此进货策略在商家不确切知道进货种类和价格的情况下被确定。蔬菜定价使用“成本加成定价法”，并且打折销售运损和品相差的蔬菜。通常还需要将需求侧和供给侧纳入制定策略的因素中。

1.2 问题要求

问题设置逐渐深入，并都与同一主题密切相关——**生鲜商超如何在不知道确切的进货种类和进货价格时做出当日补货决策，使得自身利益最大化。**其中附件表单 1 是 6 个蔬菜品类的商品信息，附件表单 2 是蔬菜类商品销售流水明细数据，附件表单 3 是每日蔬菜类商品的批发价格，附件表单 4 是不同蔬菜类商品的近期损耗率。根据附件数据信息建立数学模型，解决下述四个问题。

针对**问题一**：对蔬菜各类及单品的统计量进行统计并绘制图形，考察其时间维度上的数据分布规律。同时按照一天各时间段统计销售量，探讨各品类在一天内销量的趋势走向。检查各蔬菜分类两两之间相关性和同一类蔬菜中单品间的相关性。

针对**问题二**：根据历史销量、售价和批发价信息，以品类为单位，先探讨销售总量和成本加成定价之间的关系，再对未来一周每天的销量、售价和批发价进行预测，同时以商超收益最大化为目标，制定补货计划。

针对**问题三**：在问题二的基础上，以单品为单位，因为蔬菜商品销售空间有限，增加了单品总数和单品订购量两个限制条件，对未来 7 天的销量、售价和批发价进行预测，以商超收益最大化为目标，尽量满足市场需求，制定补货计划。

针对**问题四**：商超应该收集哪些额外的数据，从而可以优化蔬菜商品的补货和定价策略，说明意见和理由。

二、 问题分析

2.1 问题一的分析

针对第一个问题，首先计算出各蔬菜品类和蔬菜类单品的月度销量数据，接着对于 6 个蔬菜品类的月度销售数据绘制折线图和箱线图，通过图形归纳总结各蔬菜品类销售数据的分布的时序趋势、分散范围和中心位置。然后引入平均值、最大值、最小值、标准差、极差、偏度系数和峰度系数共 7 个统计量，对 6 个蔬菜品类的月度销量数据进行描述性统计。之后对于 6 个品类的销量数据，绘制散点图矩阵，概括考察不同品类间的分布规律，同时计算各品类两两之间的 Spearman 相关性系数，并绘制相关性系数矩阵和相关性系数热力图，对相关性进行深入分析。考虑到不同品类间的蔬菜单品销量差异较大，对各品类中蔬菜单品依据历史销量数据，使用系统聚类法进行聚类分析，考察在聚类数目不同时，各单品的聚类表现。同时按小时汇总各品类之间的销量数据，绘制折线图，考察不同品类蔬菜在一天中的销量变化情况之间的共性和特性。最后使用列联表卡方检验，编制列联表，考察是否退货、是否打折和蔬菜品类之间是否存在相关性。

2.2 问题二的分析

针对第二个问题，首先根据需求侧、供给侧以及历史同期的相关性，选定 2023 年 4 月至 2023 年 6 月作为预测的基本数据。之后使用回归分析考察销售总量和成本加成定价的关系。接着，根据已有的销量和售价数据，对于每个蔬菜品类，拟合出一个多项式函数。之后使用 Prophet 算法预测未来一周各品类的批发价和售价的范围，最后根据函数，考虑蔬菜类商品的损耗率，优化计算得到使得生鲜商超收益最大的品类进货策略。

2.3 问题三的分析

针对第三个问题，和第二个问题类似，首先根据需求侧、供给侧以及历史同期的相关性，选定 2023 年 4 月至 2023 年 6 月作为预测的基本数据。同时根据 6 月 24-30 日的可售品种，筛选出 7 月 1 日的所有可售单品，接着根据已有的销量和售价数据，对于每个蔬菜品类，拟合出一个多项式函数。之后使用 Prophet 算法预测未来一周的售价范围，并且使用拟合函数排除销量不足 2.5 千克的单品，最后根据剩下的单品，考虑蔬菜类商品的损耗率，优化计算得到使得生鲜商超收益最大的单品进货策略。

2.4 问题四的分析

针对第四个问题，为了更好的制定蔬菜商品的补货和定价决策，生鲜商超应该考虑从供应商、商家、消费者等方面来收集数据，以达到根据市场情况灵活调整蔬菜商品的补货和定价决策，从而更好地满足客户需求，提高竞争力，并最大程度地提高利润。本文主要分析库存量和每日的天气状况，通过绘制因果回路图研究其对蔬菜补货量和定价点的影响。

三、 模型假设

本文提出以下合理假设：

- 假设在同一天内蔬菜类单品只有一种售价
- 假设各种蔬菜单品的运损和品相情况全部保持一致
- 假设在短时间内消费者的偏好不发生改变
- 假设在短时间内货币的购买力不会发生大幅度变化

四、 符号说明

表 1 符号说明

符号	说明	单位
x	销量的原始数据	千克
x'	归一化之后的销量数据	-
X_{min}	销量数据集中的最小值	千克
$X_{\max}$	销量数据集中的最大值	千克
D_{pq}	类$\mathrm{G_p}$与$\mathrm{G_q}$之间的距离	-
d_{ij}	点X^j和点X^i之间的距离	-
O	观察频数	-
E	期望频数	-
DF	自由度	-
χ^2	卡方统计量	-

r	行的数量	-
c	列的数量	-
$Costs$	生产或购买商品的总成本	-
$Profit\ Marigin$	生产或购买商品的总成本	-
$Price$	商品的最终定价	-
q	销量	千克
p	售价	元
T_t	长期趋势	-
Y_t	整个时间序列	-
S_t	季节趋势	-
C_t	循环变动	-
R_t, ϵ_t	剩余项	-
H_t	节假日效应	-
A	该日所有蔬菜总销量	千克
$Profit$	生鲜商超的利润	元
P_i	第i个类当日销量占总销量比	-
P_{out_i}	第i类当日平均售价	元
P_{in_i}	第i类当日平均批发价	元
w_i	第i类平均运损率	-

五、 问题一的模型建立与求解

5.1 数据预处理

本题附件表单 1 给出了各蔬菜类商品的单品编号、单品名称、分类代码和各单品所属的分类名称这四种基本信息。附件表单 2 给出了从 2020 年 7 月 1 日到 2023 年 6 月 30 日中，每一天各蔬菜类商品的销售日期、扫码销售时间、单品编码、销量（单位：千克）、销售单价（单位：元/每千克）、销售类型和是否打折销售这七种销售信息。

由于在后续建模中，需要对各单品和各蔬菜商品的分类进行分布规律和相关关系的分析。首先需要对各单品和各蔬菜类商品的分类的销量按不同月份进行加总，得到从 2020 年 7 月 2023 年 6 月各单品和各蔬菜类商品的分类的月度销量。

由于后续需要分析不同蔬菜类商品销量的相关性，而不同类蔬菜商品之间重量差距较大，为使得相关性分析更加精确，对各蔬菜类商品分类月度销量数据进行归一化处理。归一化公式如下式所示。

$$x' = \frac{x - X_{min}}{X_{max} - X_{\min}}$$

对各蔬菜类商品分类月度销量的部分数据如下表所示（因数据处理量较大，故只列出部分预测结果，全部归一化结果见支撑材料）。

表 2 归一化处理后的各蔬菜类商品分类阅读销量数据（部分）

分类编码	分类名称	2020.7	2020.8	2020.9	2020.10	2020.11
1011010504	辣椒类	0.3185	0.4610	0.2715	0.2524	0.2362
1011010101	花叶类	0.5687	0.6517	0.4585	0.5466	0.5616

1011010402	水生根茎类	0.0736	0.2998	0.2709	0.5014	0.3393
1011010801	食用菌	0.2269	0.2319	0.2336	0.5937	0.6681
1011010201	花菜类	0.5380	0.6500	0.4456	0.5986	0.7465
1011010501	茄类	1.0000	0.8205	0.4494	0.5290	0.2595

5. 2 问题一的模型建立与求解

5. 2. 1 蔬菜各品类及单品销售量的分布规律

（1）商超蔬菜商品的销售情况

1）不同月份下蔬菜各品类的销售量

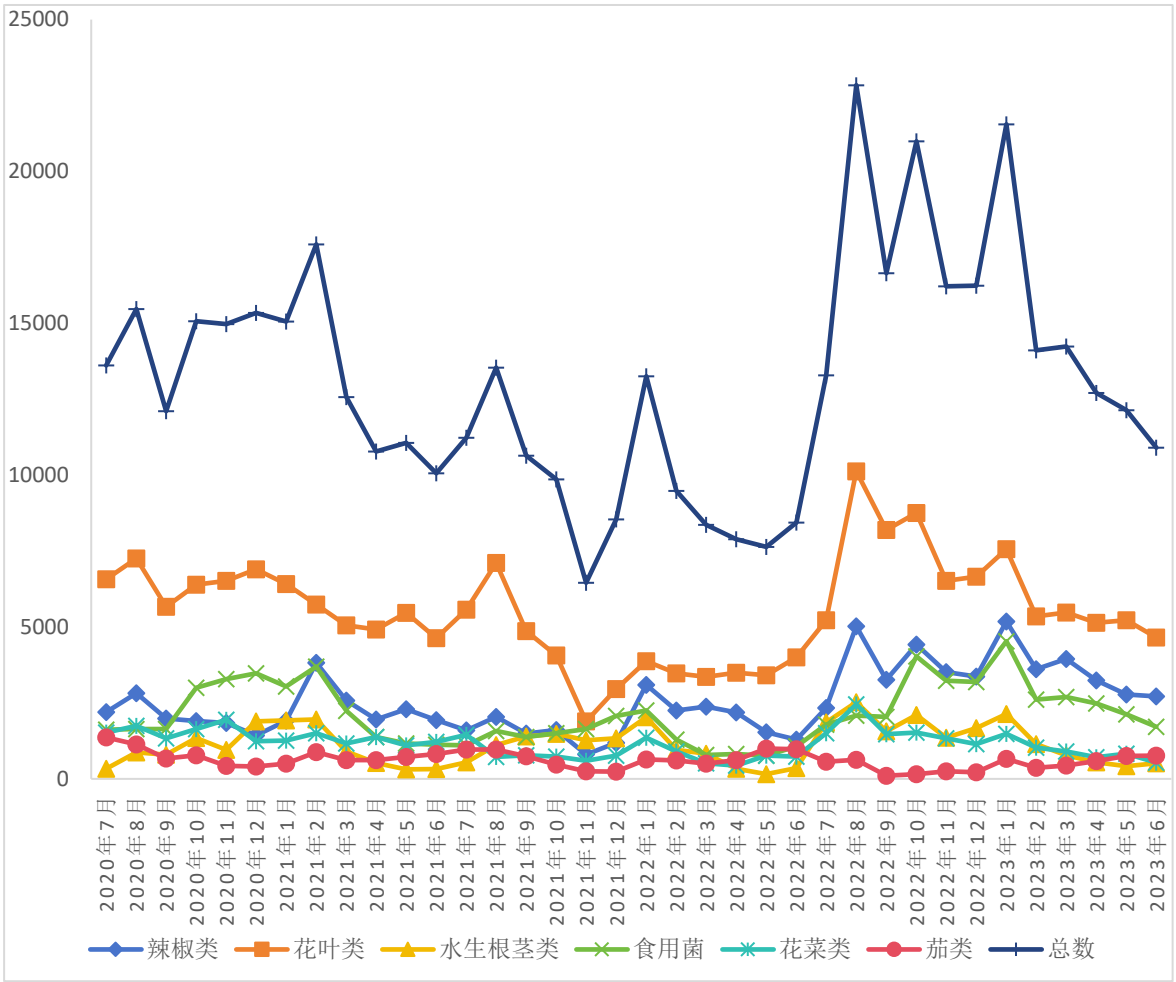


图 1 2020 年 7 月到 2023 年 6 月不同种类蔬菜的销售情况

从图中 1 可以看出，蔬菜总销量随时间的变化程度较大。受到季节因素的影响，每年 1-3 月的蔬菜总销量均呈现下降趋势，4-10 月的销售总量呈现上升趋势，10-12 月的销售总量较为平稳，变化波动较小。相较于其他季度，商家在第二、三季度需要进更多的蔬菜才能满足顾客的要求，商家要把握住蔬菜销量在不同季节的不同情况，采用科学的销售组合，合理安排蔬菜的销售空间，其中，花叶类的销量较大，它的变化是引起总销量变化的重要原因。花叶类的销量在每年第一季度呈现下降的趋势，在第二、三季度总体上升， 第四季度较为平缓。花菜类和食用菌的销售量相对于花叶类的较少，但是其销量的变化情况与花叶类的变化情况较为相似。其他三种蔬菜的销量较少，数据波动也较小，整体变化较为平稳。蔬菜的销量直接反映了顾客的需求，第一

季度的需求较少，二、三季度的需求较多、第四季度的需求较为平稳。

2）不同时刻蔬菜品类的销售量

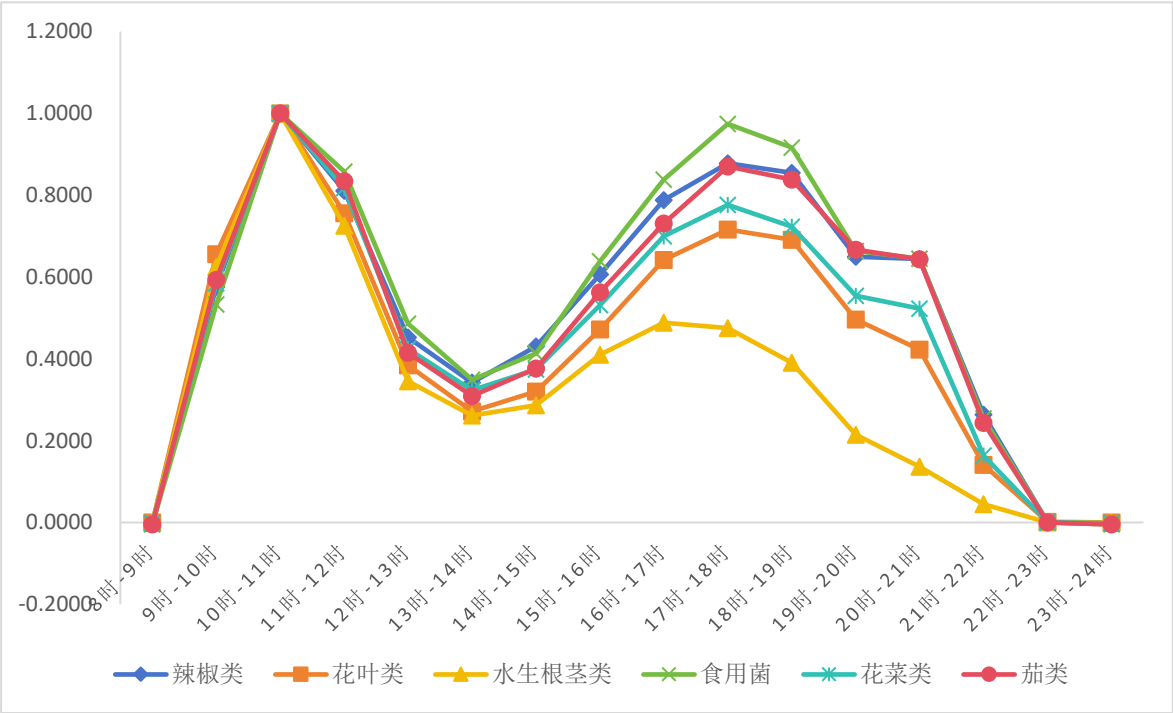


图 2 一天内蔬菜的销售情况

从图 2 中可以看出，不同种类的蔬菜的随时间的变化趋势基本相同，不同时间对蔬菜的销量影响较大。在 9 时到 11 时，蔬菜的销量快速上升，在 11 时到达一天销量的最高峰；11 时到 14 时，蔬菜的销量下降；14 时到 18 时，蔬菜销量回升；18 时到 23 时，蔬菜的销量逐渐下降。顾客对于蔬菜的需求在一天的 10 时-11 时和 17 时-18 时达到两个高峰，商家需要根据统计的销量提前做出补货。在 13 时-14 时处于需求的低峰，商家要降低补货速度，降低蔬菜的损耗。在 22 时-23 时，顾客的需求小，几乎接近 0，商家要提前清理库存，减少过夜的蔬菜量。

（2）不同品类蔬菜销售量的箱线图

箱线图可以反映出多组数据的分布特征及其分散情况，将数据的中位数、异常值和分布区间直观的呈现出来，便于进行多组数据分布情况的比较，观察样本数据的波动程度。因此，本文通过箱线图来统计各个蔬菜种类指标的数据值分布特征。箱线图绘图步骤如下：

Step1：对n个样本数据$x_1, x_2 \ldots x_n$进行从大到小排序。

Step2：找出中位数x_m，作为箱子中间的一条线。

Step3：分别计算上四分数Q_1和下四分数Q_2，并计算出箱体的长度$Q_2 - Q_1$。

Step4：分别计算出下四分位数Q_3和上四分位数Q_4作为箱子的下限和上限。

Step5：绘制出箱体、上下限、须触线，标明中位数和上下四分数，最后绘出箱线图。

超过箱子上下方的数据为异常值数据。

利用 R 语言绘制出的箱线图如图 1 所示。

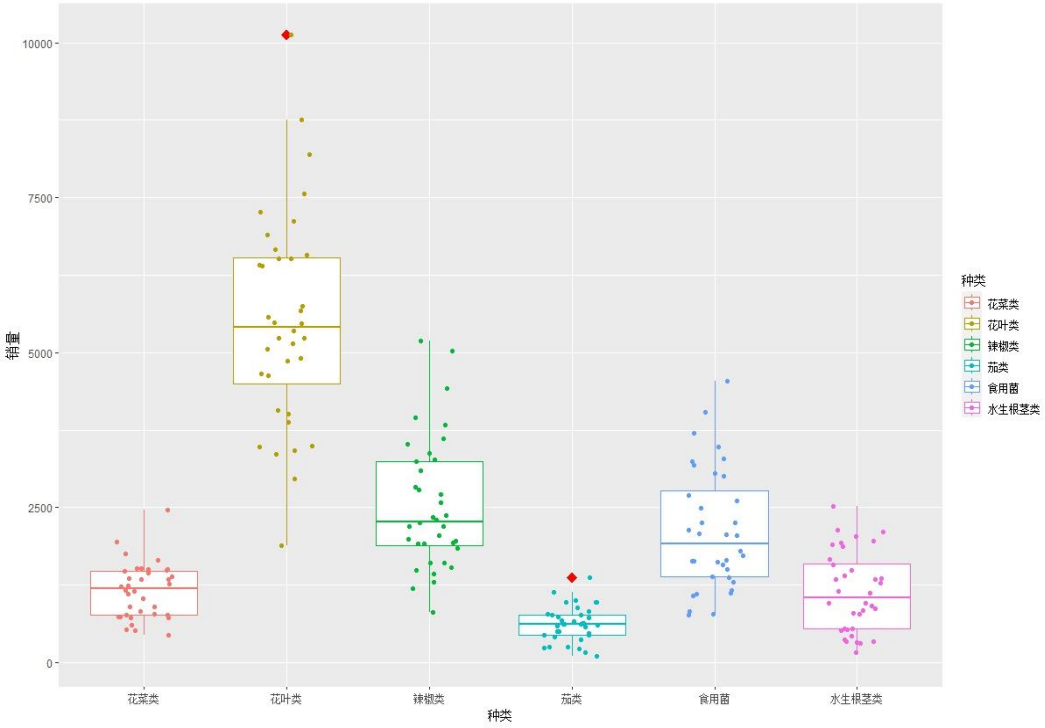


图 3 不同种类蔬菜的箱线图

由图 1 可知，不同种类的蔬菜销售情况差异较大。花叶类的中位数远超其他五种蔬菜，说明其销量较大。茄类销售数量的中位数最小，说明其销量较小。花菜类和茄类的销售数量分布更集中，数据波动较小，销售情况较为稳定。其他四种蔬菜的销售数量分布更加分散，数据波动大，销售情况不稳定。

（3）不同种类蔬菜的描述性统计

表 3 不同种蔬菜类商品月度销量描述性统计结果

	辣椒类	花叶类	水生根茎类	食用菌	花菜类	茄类
最小值	803.320	1891.080	157.830	762.230	436.830	105.320
最大值	5182.980	10131.380	2524.010	4532.330	2454.970	1365.550
极差	4379.660	8240.300	2366.180	3770.100	2018.140	1260.230
平均值	2544.128	5514.472	1127.260	2113.521	1160.179	623.104
标准差	1056.926	1721.444	645.630	969.647	445.353	284.234
偏度	0.821	0.387	0.329	0.682	0.552	0.323
峰度	0.257	0.464	-0.970	-0.265	0.511	0.164

利用 SPSS 求解六类蔬菜类商品的描述性统计量值如上表所示，其中最小值、最大值、极差、平均值和标准差的单位均为千克。

通过计算结果可以得知，辣椒类的极差较小，表明它的销量离散程度较小，较为集中，但是其标准差较大，说明该品类的销量波动较大，销售情况较为不稳定。偏度大于 0 且接近 1，说明落在均值右侧的数据偏多且数值较大，而峰度值表明指标分布相比于正态分布顶部更加尖锐或者尾部更加粗。花叶类的平均值最高，说明这类蔬菜的销量通常最多。其极差和标准差数值均较大，表示花叶类蔬菜的销量变化范围广，数据波动大。偏度大于 0，表示落在均值右侧的数据偏多。峰度值较高，说明其分布

相对尖锐。水生根茎类的平均值最低，极差较小，标准差也相对较低，表明这种蔬菜需求相对较少，数据波动小，销售情况相对稳定。其偏度接近 0.33，表示其销售数量为右偏分布。峰度值为负，表明其销售数量分布较平坦，没有明显的峰值。食用菌的平均值略低于花叶类，但高于其他蔬菜种类。极差较大，标准差较高，表明食用菌的销量波动较大。偏度大于 0 接近 0.68，说明落在均值右侧的数据偏多，存在较大的值。峰度值为负，表明指标分布其相比于正态分布顶部更加平坦或者尾部更加细。花菜类的数据表现出相对稳定的特点，极差较小，标准差最低，数据波动小。偏度接近 0.55，属于右偏分布，存在较大的值。峰度值较高，表明分布相对陡峭。茄类的极差和标准差都最小，数据波动较小，销售情况最稳定。偏度大于 0，说明落在均值右侧的数据偏多。峰度值为正，分布相对陡峭。

总体而言，不同种类的蔬菜在数量分布上具有各自的特点。花叶类的平均值、极差和标准差数值均最大，说明该品种的蔬菜销售数量最多、需求最大，但是其销售情况并不稳定，数据波动较大。茄类的平均值、极差和标准差均最小，说明其销售数量少，销售情况稳定。其他种类的蔬菜销售数量的平均值、极差和标准差数值都相对较小，说明这些品种的蔬菜销售情况较为稳定。六种蔬菜的偏度均大于 0，说明其销售数量落在对应均值右侧的数据偏多，其中辣椒类为 0.821，说明其销售数量数值偏大。水生根茎类和食用菌的峰度均为负，表明指标分布其相比于正态分布顶部更加平坦或者尾部更加细。其他四类均为正，表明指标分布相比于正态分布顶部更加尖锐或者尾部更加粗。

5.2.2 不同品类蔬菜销售量之间的相互关系

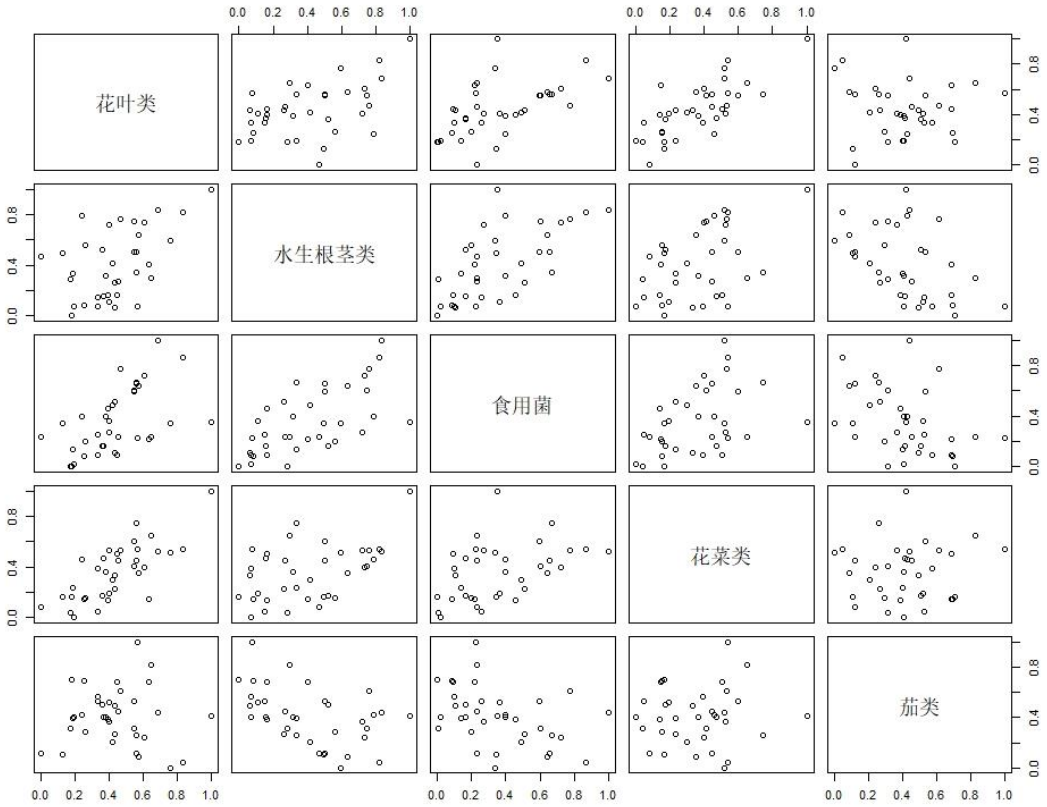


图 4 各解释变量和被解释变量的散点图矩阵

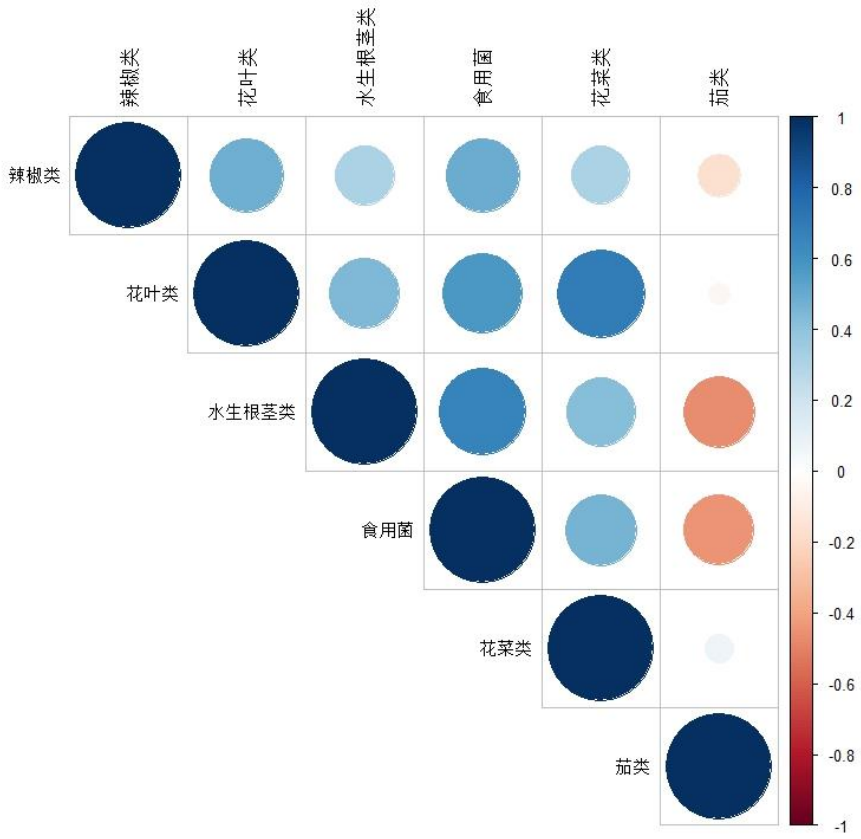


图 5 各解释变量和被解释变量的相关系数热力图

如图所示，辣椒类和花叶类、食用菌之间存在较强的正相关性，和水生根茎类、花菜类之间存在较弱的正相关性。花叶类与花菜类、食用菌之间存在较强的正相关性，与水生根茎类存在较弱的正相关性。水生根茎类和食用菌、花菜类之间存在较强的正相关性，和茄类之间存在较强的负相关性。食用菌与花菜类存在较强的正相关性，与茄类存在较强的负相关性。其他数据之间则相关性不强。

表 4 各蔬菜分类的斯皮尔曼相关性系数矩阵（表示在 0.01 的水平上显著）**

蔬菜分类	辣椒类	花叶类	水生根茎类	食用菌	花菜类	茄类
辣椒类	1.00	0.488**	0.316	0.490**	0.310	−0.170
花叶类	0.488**	1.00	0.448**	0.578**	0.695**	−0.043
水生根茎类	0.316	0.448**	1.00	0.669**	0.427**	−0.467**
食用菌	0.490**	0.578**	0.669**	1.00	0.005	0.006
花菜类	0.310	0.695**	0.427**	0.005	1.00	0.076
茄类	−0.170	−0.043	−0.467**	0.006	0.076	1.00

相关性系数衡量了两个变量之间的关联程度，其值范围从-1 到 1，正值表示正相关，负值表示负相关，0 表示无关。相关性系数的绝对值越接近于 1 表明两者直接的关系越密切。表 1 展示了不同蔬菜分类之间的相关性系数。

从表中我们可以得知：辣椒类与花叶类、食用菌在 1%的水平上显著正相关，与其他四类的相关性较小。花叶类与水生根茎类、食用菌和花菜类在 1%的水平上显著正相关，与茄类的相关性较小。其中，花叶类与花菜类的相关性系数为 0.695**，说明两者之间的相关性很强，对彼此的销售情况影响较大。水生根茎类与食用菌、花菜类在 1%的水平上显著正相关，与茄类呈显著负相关。食用菌类和花菜类、茄类无显著相关性，花菜类和茄类无显著相关性。其中，值得注意的是，茄类与其他蔬菜种类之

间多呈现负相关性，其销售数量的增加在一定程度上会降低其他蔬菜的销售量。

5.2.3 蔬菜单品销售量的分布规律

在分析蔬菜类商品单品相关性时，考虑到不同分类蔬菜商品之间重量差异较大，为找出销量将存在相似性的不同单品，将在同一个蔬菜类中使用系统聚类法，把具有相似销量特征的蔬菜单品并入同一个类。

聚类(Clustering)就是按照某个特定标准把一个数据集分割成不同的类或簇，最后的结果是希望同类之间的差异性尽可能小，使得不同类之间的差异性尽可能大；不同的类具有能够表达异于其他类的指标。目前常用的聚类方法有：系统聚类法(Hierarchical Clustering)、K-means 聚类和 DBSCAN 聚类等。其中系统聚类的原理是将每个变量看作单独的一类，然后根据不同点之间的距离逐步合并成大类，直到所有变量都能被归为同一类。一般使用树状图和聚类谱系图表示系统聚类结果。

系统聚类的分析步骤如下（以最短距离法为例）：

Step1.确定点与点之间的距离。在进行系统聚类之前，首先需要定义点与点之间的距离。常用的点间距离定义方法有 8 种，分别是最短距离发、最长距离法、中心距离法、重心法、类平均法、可变类平均法、可变法和离差平方和法。不同方法之间主要差异是确定类间距离的计算方法不同。

最短距离法定义类与类间距离的方式如下式：

$$D_{pq} = \min_{X^i \in G_k, X^j \in G_q} d_{ij}$$

其中，d_{ij}表示点X^i和X^j之间的距离，D_{pq}表示类G_p与G_q之间的距离。

设之后类G_p和G_q合并成一个新的类记作G_r，则G_r与另外一个类G_k之间的距离为下式(2)：

$$D_{kr} = \min_{X^i \in G_k, X^j \in G_r} d_{ij} = \min\left\{ \min_{X^i \in G_k, X^j \in G_p} d_{ij}, \min_{X^i \in G_k, X^j \in G_q} d_{ij} \right\} = \min\{D_{kp}, D_{kq}\}$$

类似地，最长距离法定义类与类之间的距离为两类最远样品的距离；重心法定义类与类之间的距离为两个类的重心之间的距离，其他方法不再详述。

Step2.开始聚类。确定距离最小的两个点之间的距离，设为D_{pq}，接着将G_p和G_q合并为一个新的类，记作G_r，即$G_r = \{G_p, G_q\}$。

Step3.按照上文式(1)的公式计算新的类与其他类的距离。

Step4.重复 2、3 两步，直到所有元素并成一类为止。如果某一步距离最小的元素不一个，则对应这些最小元素的类可以同时合并。

使用 R 语言对不同蔬菜类商品的分类进行系统聚类，可得如下结果：

通过对花叶类蔬菜的树状图（见附录 9）进行分析我们可以得到以下结果：娃娃菜、保康高山大白菜、云南生菜、云南油麦菜、芝麻苋菜、菠菜与其他花叶类蔬菜销量数据之间有明显的不同；将所有花叶类蔬菜分为 3 类时，这六种蔬菜聚为一类；将所有花叶类蔬菜分为 4 类时，这六种蔬菜并为一类；将所有花叶类蔬菜聚为 4 类时，这六种蔬菜同样并为一类。联系现实生活，这六种蔬菜具有季节性的生长周期，它们可能在相似的时间内进入市场；并且这些蔬菜可能在市场上具有类似的市场定位。它们通常都被归类为绿叶蔬菜，富含维生素、矿物质和纤维，适合制作沙拉、炒菜、汤

等菜肴；同时这些蔬菜通常都受到健康意识高、注重膳食多样性的消费者群体的青睐。因此，它们可能在相似的消费者群体中有一定的重叠，这可能导致它们在销量数据中表现出相似的销售趋势。

通过对花菜类蔬菜的树状图（见附录 9）进行分析我们可以得到以下结果：将所有花菜类蔬菜聚为 4 类时，由于紫白菜(1)和紫白菜(2)是同一种蔬菜单品只是产地不同，所以二者被聚为同一类；将所有花菜类蔬菜分为 3 类时，青梗散花和枝江青梗散花并没有被聚为一个类，联系现实，可以推测消费者认为青梗散花和枝江青梗散花在多方面存在巨大不同。

通过对水生根茎类蔬菜的树状图（见附录 9）进行分析我们可以得到以下结果：将所有水生根茎类蔬菜聚为 3、4、5 类时，不同产地的净藕都没有被聚为同一类，联系销量数据，据此推测不同产地的净藕上市时间和口感之间都存在差异；同时将所有水生根茎类蔬菜分为 3、4、5 类时，洪湖莲藕(粉藕)和洪湖莲藕(脆藕)也没有被聚为一类，联系销量数据推测二者上市时间不同。

通过对茄类蔬菜的树状图（见附录 9）进行分析我们可以得到以下结果：将所有茄类蔬菜聚为 2、3、4、5 类时，紫茄子(1)和紫茄子(2)均没有聚为一类，联系销量数据，推测这和生鲜商超的进货策略有关，生鲜商超可能仅在 2022 年 8 到同年 10 月及 2023 年 5 月到同年 6 月少量进货紫茄子(1)，在其他月份均只进货紫茄子(2)。

通过对辣椒类蔬菜的树状图（见附录 9）进行分析我们可以得到以下结果：将所有辣椒类蔬菜聚为 2、3、4、5 类时，水果辣椒(橙色)均是单独被聚为一类，联系销量数据发现其只在 2022 年 2 月有过销售；同时将所有辣椒类蔬菜分为 4、5 类时，小米椒、小邹皮和螺丝椒均被聚为同一类，联系现实发现这三种辣椒都是辣度较高的辣椒，贴合聚类结果。

通过对食用菌蔬菜的树状图（见附录 9）进行分析我们可以得到以下结果：将所有食用菌蔬菜聚为 2 类以上时，金针菇（盒）总是单独聚为一类，联系销量数据可以得知，金针菇（盒）生鲜商超从 2022 年 1 月才开始进货销售的一类蔬菜单品，也是唯一一类从 2022 年 1 月才开始进货销售的蔬菜单品，贴合聚类结果。

5.2.4 列联表独立性检验蔬菜类商品的分类是否与销售类型和是否打折相关

为了检验蔬菜类商品的分类是否与销售类型和是否打折相关，将采用列联表独立性检验的方式对数据进行分析。

列联表卡方检验的目标是确定观察到的频数与期望频数之间的差异是否足够大，以至于我们可以得出两个变量之间是否存在显著关联。其原理基于以下假设和思想：

1.零假设H_0：假定两个变量之间是独立的，即它们之间没有关联性，事件的发生与另一个变量无关。

2.备择假设H_1：假定两个变量之间存在关联性，事件的发生与另一个变量有关。

列联表卡方检验的原理如下：

1. 建立列联表：首先，根据收集到的数据建立一个列联表，该表汇总了两个分类变量之间的交叉频数。列联表的行代表一个分类变量的不同水平，列代表另一个分类变量的不同水平。

2.计算期望频数：在零假设成立的情况下，计算出每个单元格中的期望频数。期望频数是指在独立性假设下，每个单元格中事件的预期发生次数。通常，期望频数的计算基于总体事件的概率和每个分类变量水平的边际频数。

3. 计算卡方统计量：卡方统计量（Chi-Square statistic）用于比较观察频数和期望频数之间的差异，计算方式为将每个单元格中的观察频数与期望频数之差的平方，然后除以期望频数，再对所有单元格的结果求和。卡方统计量的计算公式如下：

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

其中，χ^2是卡方统计量，O是观察频数，E是期望频数。

4.确定自由度：卡方检验的自由度（degrees of freedom）取决于列联表的维度。通常，自由度有如下计算公式：

$$DF = (r - 1) \times (c - 1)$$

其中，r是行的数量，c是列的数量。

5. 比较计算的卡方统计量与临界卡方值：如果计算的卡方统计量大于临界卡方值，则拒绝零假设，即认为两个变量之间存在关联性。如果计算的卡方统计量小于临界卡方值，则无法拒绝零假设，即认为两个变量之间独立。

分别对不同蔬菜类商品分类和销售类型、不同蔬菜类商品分类和是否打折在 2020 年 7 月到 2023 年 6 月发生的次数汇总，得到了如下两个列联表。

表 5 不同蔬菜类商品销售类型列联表

	正常销售	退款
花叶类	331789	46
花菜类	86524	17
水生根茎类	58613	102
茄类	44881	34
辣椒类	207894	83
食用菌	148341	179

表 6 不同蔬菜类商品是否打折列联表

	正常销售	打折销售
花叶类	315286	16682
花菜类	81924	4646
水生根茎类	52856	5791
茄类	43740	1158
辣椒类	200603	7393
食用菌	136728	11696

使用 R 软件进行列联表独立性卡方检验，得到如下结果：

表 7 列联表独立性卡方检验的结果

	$\chi^2 - value$	df	p
销售类型	418.69	5	$< 2.2e-16$
是否打折	6272.3	5	$< 2.2e-16$

由于两个列联表检验的p值都小于$2.2e-16$，因此拒绝原假设，即认为商品种类和商品销售类型之间存在相关性，商品种类和是否打折销售之间存在相关性。

六、 问题二的模型建立与求解

6.1 “成本加成定价”定价法：

成本加成定价法是一种相对传统的定价法，其起源最早可以追溯到工业化时期的早期，并且在工业革命后的 19 世纪末期和 20 世纪前叶变得流行，主要在制造业和工程领域广泛盛行。

成本加成定价法的定价公式如下式。

$$\text{Costs} + \text{Profit Marigin} = \text{Price}$$

其中，Costs指的是生产或购买商品的总成本（包括直接人工成本、制造成本、运输成本等），Profit Margin指的是想要获得的边际利润，Price 指的是商品的最终定价。

成本加成定价法的有点包括计算简单、成本覆盖、稳健盈利和适用范围广等；其缺点有其忽略市场需求、不适用于高度竞争的市场、缺乏灵活性和没有通过扩大收入或调整来实现利润最大化的动力等。所以，成本加成定价法适用于制造业和生产业务、固定成本高的行业、低竞争度市场、新产品定价、定制产品或服务、合同定价和利润稳健性要求等情况；但是它不适用于竞争激烈的市场、差异化产品或服务、高度变动的成本结构等情况。

6.2 蔬菜品类的销售总量与成本加成定价的关系

为了分析各蔬菜品类的销售总量与成本加成定价的关系，我们使用各蔬菜品类的批发价作为成本加成定价的代理变量，建立线性回归模型。使用 OLS 方法，以 2023 年 4 月至 6 月每天各蔬菜品类的销量作为被解释变量，以 2023 年 4 月至 6 月每天各蔬菜品类的批发价作为解释变量，设销量为q，批发价为p，求解参数，可以得到以下结果。

表 8 各蔬菜品类销量和批发价的 OLS 结果

蔬菜品类	回归方程	拟合优度
茄类	$q = -1.6024p + 31.151$	0.0167
水生根茎类	$q = -0.2764p + 18.579$	0.0107
花菜类	$q = -2.6449p + 44.243$	0.1293
花叶类	$q = -12.248p + 205.45$	0.0066
辣椒类	$q = -5.8871p + 118.63$	0.0067
食用菌	$q = -0.6292p + 71.838$	0.0001

综上可以发现，所有的批发价和销量之间均呈现负相关关系，但是拟合优度R^2都不超过0.15，处于比较低的水平。这说明各蔬菜品类的销售总量与成本加成定价的关系并不显著，成本加成定价法的核心因素——成本不会对于各蔬菜类销量造成显著的影响。

6.3 Prophet 方法

Prophet 方法是由 Facebook 研发的时间序列预测方法，其主要目标是为了准确预测那些呈现出季节性和趋势性模式的时间序列数据。Prophet 方法具备一系列显著的特点，其中包括出色的易用性、高度自动化的特性以及对各类时间序列数据的广泛适用性。

在传统的时间序列分析中，时间序列的分解（Decomposition of Time Series），时间序列可用多种模型进行分解，常见的有加法模型、乘法模型和加乘混合模型。将时间序列Y_t分成几个部分，分别是长期趋势(T_t)、季节趋势(S_t)、循环变动(C_t)、和剩余项(R_t)。

对于加法模型而言，设所有$t > 0$，都有以下分解式：

$$Y_t = T_t + S_t + C_t + R_t$$

对于乘法模型而言，设所有$t > 0$，都有以下分解式：

$$Y_t = T_t \times S_t \times C_t \times R_t$$

对于加乘混合模型而言，设所有$t > 0$，都有以下分解式（不唯一）：

$$Y_t = T_t + S_t + C_t \times R_t$$

在 Prophet 方法中，作者同时考虑了季节项、趋势项、节假日的效应和剩余项四

个效应，即下式所示：

$$Y(t) = T(t) + S(t) + H(t) + \epsilon_t$$

其中，$T(t)$表示趋势项，即时间序列在非周期上的变化趋势，$S(t)$表示季节项，一般来说以月份以及年份为单位；$H(t)$表示节假日项，表示在当天是否是节假日；ϵ_t表示误差项或者剩余项。Prophet 算法通过将这四项累加得到了时间序列的预测值。

6.4 蔬菜品类售价的预测

使用 Python 将附件 2、3 中的数据进行整理，从供给两侧进行考虑，4—6 月份与题目中要求的 7 月 1 日至 7 月 7 日同为蔬菜供应品种较为丰富的月份；同时，使用一个季度的时间序列数据进行预测既能防止数据被季节项所错误预测，又能保持预测数据保持原有数据的长期趋势。综上，决定使用 2023 年 4 月—6 月一共 91 天各品类的销售价格和批发价格作为原始的预测项。绘制如下示意图。

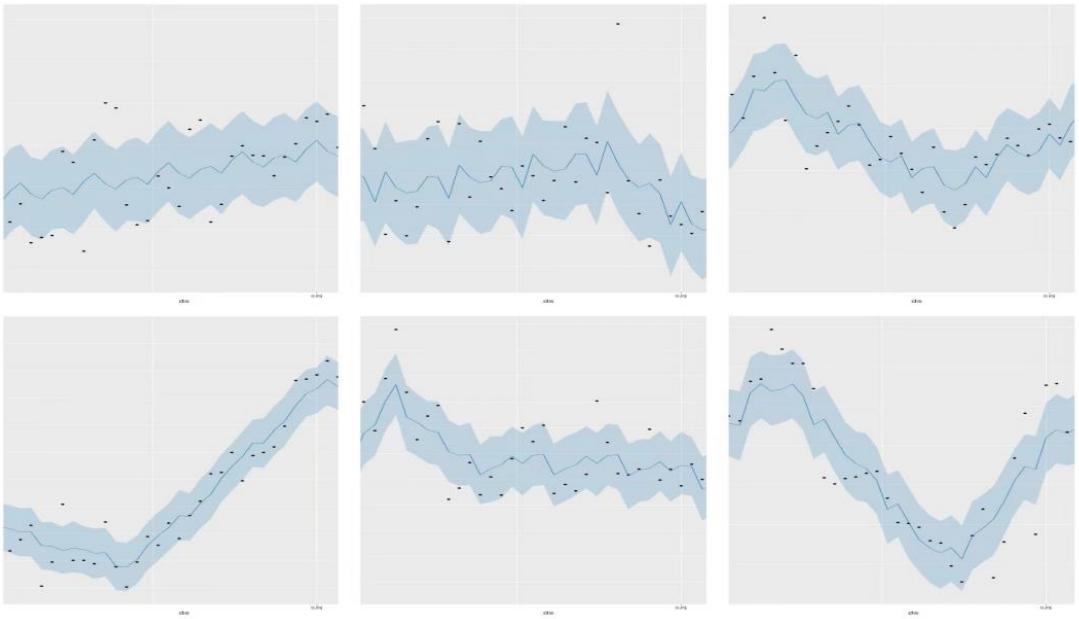


图 6 六个蔬菜品类售价的预测图象

6.5 各蔬菜品类未来一种的补货总量和定价策略

6.5.1 模型建立逻辑

（1）目标：求每日利润最大时对应的售价和进货量

（2）实现方式：将每日总利润表示为 6 个品类的销售价格的函数；为每个品类的价格和销量建立线性回归模型，通过价格等因素预测每日销售量；将每个品类包含单品的运损率按销售数量取加权平均，作为该品类的运损率；使用 Prophet 算法，基于 2023 年 4-6 月每日每个品类的销售价格预测未来 7 天定价的合理范围、基于该时段每日每个品类的批发价格预测未来 7 天批发价格的值，将该定价合理范围作为约束条件、将该批发价格结合运损率进行成本加成定价，规划求解每日总利润最大时对应的销售价格以及对应的销售量，即进货减去运损量；

（3）建立每个品类销售量关于价格的线性回归模型：注意到每日总销量波动较大且存在明显的以 7 天为周期的周期性（可以较为准确地预测未来 7 天的总销量）、无论是单个品类还是总体的销量都难以和销售价格之间建立良好的线性回归模型；又注意到单个品类的每日销量占当天销量的比例和该品类当天的价格具有较为显著的线性相关性——建立销量占比关于价格的线性回归模型，将每日销量占比乘以 Prophet

预测的每日总销量作为当日该品类销量的预期值，从而建立销售量关于价格的线性回归模型。

6.5.2 模型求解

每日利润表示为下式：

$$Profit = \sum_{i=1}^{6} A * P_i * \left(P_{out_i} - \frac{P_{in_i}}{1 - w_i} \right)$$

其中，A为该日所有蔬菜总销量；P_i为第i个品类当日销量占总销量比；P_{out_i}为第i个品类当日平均售价；P_{in_i}为第i个品类当日进货价格；w_i为第i个品类的平均运损率；【对应关系为：1，茄类；2，水生根茎类；3，花菜类；4，花叶类；5，辣椒类；6，食用菌】

约束条件(s.t.)如下式所示：

$$\sum_{i=1}^{6} P_i = 1$$

$$\min\left(\widehat{P_{out_i}}\right) \leq P_{out_i} \leq \max\left(\widehat{P_{out_i}}\right)$$

销量占比关于售价的回归方程如下式：

$$\hat{P}_1 = \frac{-0.660068299516679 * P_{out_1} + 11.201977190390584}{100}$$

$$\hat{P}_2 = \frac{-2.192659352557455 * P_{out_2} + 52.907948136773435}{100}$$

$$\hat{P}_3 = \frac{-0.07554540537148258 * P_{out_3} + 4.85875861263871}{100}$$

$$\hat{P}_4 = \frac{-0.4244825299711383 * P_{out_4} + 10.722063204474223}{100}$$

$$\hat{P}_5 = \frac{-1.6721096166348721 * P_{out_5} + 35.13549201815904}{100}$$

$$\hat{P}_6 = \frac{-0.6524530882325468 * P_{out_6} + 21.052092209314672}{100}$$

基于 Prophet 的预测结果，得到各品类进货价格的预测值如下表所示。从表中可以看出，茄类保持在每千克 4.20 元左右，水生根茎类的价格在每千克 13.00 元上下浮动，花菜类的价格保持在每千克 7.00 元以上，花叶类、辣椒类、食用菌的价格保持在每千克 3.00-4.00 元之间。每种蔬菜的进货价格在一周的价格波动变化较少，均符合实际情况，可信度较高。

表 9 各品类蔬菜进货价格预测值（单位：元/kg）

价格	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
P_{in_1}	4.26	4.27	4.31	4.27	4.01	4.20	4.07
P_{in_2}	12.67	13.15	13.35	13.07	13.75	14.21	13.76
P_{in_3}	7.63	7.53	7.38	7.35	7.24	7.11	7.03
P_{in_4}	3.27	3.30	3.22	3.16	3.15	3.22	3.11
P_{in_5}	3,07	3,10	3,12	3,19	3,09	3,15	3,14
P_{in_6}	3.67	3.68	3.79	3.78	3.60	3.87	3.76

各品类定价的合理范围预测如下表所示。其中花菜类和水生根茎类的定价均位于每千克 10.00 元以上，其他种类蔬菜的价格在每千克 4.00-8.00 元之间。每种蔬菜的定价下界均保持在预测的进货价格之上，定价上界也符合市场中该类蔬菜可能达到的最

高价格，符合实际情况，具有较高可信度。

表 10 各品类蔬菜定价的合理范围预测（单位：元/kg）

品类	界限	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
花菜类	下界	11,64	11,49	11,55	11,35	11,13	10,81	10,74
花菜类	上界	13,25	13,13	13,02	13,03	12,79	12,67	12,45
花叶类	下界	4,63	4,61	4,52	4,37	4,39	4,48	4,34
花叶类	上界	5,21	5,23	5,14	5,01	5,04	5,16	5,04
辣椒类	下界	5,04	5,2	5,25	5,3	5,3	5,27	5,24
辣椒类	上界	5,99	6,19	6,16	6,27	6,25	6,26	6,33
茄类	下界	6,39	6,3	6,21	6,16	6,07	6,4	6,16
茄类	上界	7,67	7,52	7,49	7,45	7,49	7,91	7,8
食用菌	下界	4,88	4,93	5,05	5,06	4,85	5,19	4,98
食用菌	上界	6,19	6,23	6,34	6,32	6,14	6,51	6,36
水生根茎类	下界	13,37	13,77	13,86	13,07	14,08	14,86	14,17
水生根茎类	上界	18,33	19,2	19,23	18,76	19,5	20,37	19,75

各品类运损率平均值如下表所示。受到运输距离和储存环境的影响，辣椒类、花叶类、水生根茎类的运损率较高，达到 10.53%、9.70%、8.14%，因此商家要注意在满足辣椒类、花叶类、水生根茎类需求量的前提下适当提高它们的的进货量，避免因运损过高导致库存不足。

表 11 各品类蔬菜运损率平均值

品类	茄类	水生根茎类	花菜类	花叶类	辣椒类	食用菌
运损率(%)	7,49	8,14	5,50	9,70	10,53	7,22

结合以上模型假定和数据基础，使用非线性规划模型求解，可得每日每个品类的定价和进货量应为下表所示。水生根茎的定价最高，保持在每千克 19.00 元左右，其进货数量最少，均在 20 千克以下。花叶类和辣椒类的定价较低，但是其进货量较高。

表 12 各蔬菜品类的定价和进货量

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	
品类	售价(元/kg)	批发量（千克）	售价(元/kg)	批发量（千克）	售价(元/kg)	批发量（千克）	售价(元/kg)	批发量（千克）	售价(元/kg)	批发量（千克）	售价(元/kg)	批发量（千克）	售价(元/kg)	批发量（千克）
茄类	6.39	35.40	6.30	33.43	6.21	20.57	6.16	21.46	6.07	21.55	6.40	18.25	6.16	22.64
水生根茎类	18.33	17.73	19.20	16.29	19.23	9.93	18.76	10.42	19.50	10.21	20.37	8.75	19.75	10.75
花菜类	11.64	28.68	11.49	27.16	11.55	16.50	11.35	17.38	11.13	17.58	10.81	15.71	10.74	19.14
花叶类	5,87	207.88	5.23	201.52	5.14	123.56	5.01	129.17	5.04	128.42	5.16	111.47	5.04	136.03
辣椒类	5.04	139.95	5.90	124.05	5.99	75.23	6.26	76.71	6.26	76.36	5.96	68.08	6.32	80.60
食用菌	6.19	85.97	6.23	80.40	6.34	48.85	6.32	50.76	6.36	50.47	6.51	43.83	6.36	53.46

七、 问题三的模型建立与求解

7.1 数据选取

因为题目三中提到根据 2023 年 6 月 24 日—6 月 30 日的可售品种，来制定 7 月 1 日的补货量和定价策略，所以通过汇总附件表格 2 的销售数据，排除 2023 年 6 月 24 日—6 月 30 日中生鲜商超没有出现的蔬菜单品；同时为了使得各单品订购量满足最小陈列量 2.5 千克的要求，也排除了在 2023 年 4 月—2023 年 6 月间最大销售量不足 2.5 千克的蔬菜单品，最终剩余 41 个蔬菜单品可供补货和定价策略选用。

7.2 蔬菜单品售价区间和批发价预测

表 13　41 种蔬菜单品 7 月 1 日的售价预测区间和批发价预测值（单位：元/kg）

单品名称	售价预测值下界	售价预测值上界	批发价预测值	单品名称	售价预测值下界	售价预测值上界	批发价预测值
小皱皮(份)	1.88	2.86	1.43	金针菇(盒)	1.64	2.08	1.42
竹叶菜	0.83	4.83	1.36	小青菜(1)	4.28	5.45	2.64
苋菜	2.61	5.28	2.05	木耳菜	0.69	9.45	2.74
云南生菜(份)	4.16	4.75	3.53	双孢菇(盒)	4.78	5.43	3.32
上海青	7.68	9.33	4.15	高瓜(1)	7.77	16.15	11.86
西峡花菇(1)	24.46	25.4	15.63	洪湖藕带	7.52	30.2	18.03
圆茄子(2)	4.43	11.7	4.28	青茄子(1)	3.98	8.93	4.1
芜湖青椒(1)	4.82	5.72	3.39	螺丝椒(份)	1.78	5.14	3.28
云南油麦菜(份)	3.88	4.56	2.95	螺丝椒	5.78	14.76	7.5
海鲜菇(包)	2.5	3.34	1.91	蟹味菇与白玉菇双拼(盒)	0.01	7.04	3.15
西兰花	11.47	12.76	7.31	菱角	9.15	24.54	9.2
净藕(1)	14.42	18.4	13.88	姜蒜小米椒组合装(小份)	3.91	5.59	2.45
小米椒(份)	4.66	7.08	1.96	七彩椒(2)	9.93	25.18	13.26
红薯尖	1.42	9.41	2.41	菠菜(份)	3.23	5.37	4.39
紫茄子(2)	5.75	6.26	3.57	红莲藕带	7.03	14.63	5.47
枝江	1.35	10.21	9.8	青红杭椒组	1.65	7.38	3.44

青梗散花				合装(份)			
奶白菜	3.3	5.67	2.42	青线椒(份)	0.01	4.38	2.43
菜心	0.01	6.63	4.62	云南生菜	2.1	9.72	5.58
娃娃菜	6.07	6.55	4.64	白玉菇(袋)	0.01	6.78	3.43
虫草花(份)	1.47	5.03	2.64	木耳菜(份)	0.58	4.23	1.42
长线茄	8.73	13.24	6.67				

整理附件表格 2、3，得到 2023 年 4 月—2023 年 6 月各单品的批发价和销售价格数据，并且使用 Prophet 算法，预测得到 7 月 1 日各单品销售价格的区间最大值、区间最小值和批发价格的预测值，如上表所示。其中，售价预测值上界最高的是洪湖藕带，达到每千克 30.2 元；售价预测值下界最低的有菜心、蟹味菇与白玉菇双拼和青线椒，仅有每千克 0.01 元。每种单品的售价区间基本符合目前市场上蔬菜的实际价格，且区间的间隔较小，数据可靠性较高。

7.3 售价和进货量的确认

7.3.1 模型逻辑

（1）目标：确定进货单品及各自的售价和进货量使得总利润最大

（2）定价和补货决策：参考问题二，仍然将 7 月 1 日预测利润表示为 41 个单品的销售价格的函数；单品的运损率取近期盘货所得运损率的值；7 月 1 日总销量预测和问题二相同；建立每个单品的日销售量占比关于价格的回归模型，将占比预测乘以日总销量预测得到该单品 7 月 1 日销售量关于其价格的函数。

7.3.2 模型求解

每日利润表示为：

$$Profit = \sum_{i=1}^{41} A * P_i * \left(P_{out_i} - \frac{P_{in_i}}{1 - w_i} \right)$$

其中各参数的意义参考问题二中的解释；

约束条件：

$$\sum_{i=1}^{41} P_i = 1$$

$$\min\left(\widehat{P_{out_i}}\right) \le P_{out_i} \le \max\left(\widehat{P_{out_i}}\right)$$

考虑到变量数众多，此处不具体罗列回归方程，其统一形式为：

$$\widehat{P_i} = \widehat{b_i} * P_{out_i} + \widehat{a_i}$$

使用 Prophet 预测的各单品售价区间和批发价的值已在本部分表 1 中给出。每日总销量的预测值和问题二中所列相同，7 月 1 日预测为 468，82kg。

每个单品的损耗率如下图所示。其中损耗率最高的是高瓜，高达 29.25%，损耗率最低的是海鲜菇，几乎为 0.

表 14 每个单品的损耗率

单品名称	损耗率(%)	单品名称	损耗率(%)
菱角	9.61	虫草花(份)	9.43
洪湖藕带	24.05	云南油麦菜(份)	9.43
云南生菜	15.25	枝江青梗散花	9.43
长线茄	6.9	小青菜(1)	10.33
菜心	13.7	小皱皮(份)	9.43
青线椒(份)	9.43	竹叶菜	13.62
木耳菜(份)	9.43	娃娃菜	2.48
青红杭椒组合装(份)	9.43	双孢菇(盒)	0.2
奶白菜	15.68	姜蒜小米椒组合装(小份)	9.43
红莲藕带	16.63	蟹味菇与白玉菇双拼(盒)	0.84
海鲜菇(包)	0	芜湖青椒(1)	5.7
螺丝椒	10.18	净藕(1)	5.54
白玉菇(袋)	6.57	上海青	14.43
青茄子(1)	5.01	小米椒(份)	9.43
红薯尖	8.42	紫茄子(2)	6.07
木耳菜	7.61	螺丝椒(份)	9.43
圆茄子(2)	6.71	西兰花	9.26
菠菜(份)	9.43	高瓜(1)	29.25
金针菇(盒)	0.45	西峡花菇(1)	10.8
云南生菜(份)	9.43	七彩椒(2)	9.43
苋菜	18.52	-	-

因此，总利润的期望可以表示为：

$$\widehat{Profit} = A * \sum_{i=1}^{41} \left(\hat{b_i} * P_{out_i} + \hat{a_i}\right) * \left(P_{out_i} - \frac{P_{in_i}}{1 - w_i}\right)$$

结合总利润的期望表达式和上述各项约束条件，使用非线性规划求解可以得出最优解，该解包含了所有蔬菜品类，满足品类齐全的条件。因此，为使得商超蔬菜种类尽可能齐全的同时利润期望值最大化，应该选择的单品以及各单品的进货量和成本加成定价的价格应为：

表 15 各单品的进货量和成本加成定价的价格

名称	售价(元)	数量(kg)	名称	售价(元)	数量(kg)
奶白菜	4.49	19.47	竹叶菜	2.83	23.66
海鲜菇(包)	2.92	17.72	娃娃菜	6.31	16.28
螺丝椒	10.27	13.52	双孢菇(盒)	5.11	18.59

青茄子(1)	6.46	3.04	姜蒜小米椒组合装(小份)	4.75	12.14
红薯尖	5.42	9.54	芜湖青椒(1)	5.27	25.36
木耳菜	5.07	8.83	净藕(1)	16.41	9.04
菠菜(份)	4.3	25.54	上海青	8.51	8.91
金针菇(盒)	1.86	29.78	小米椒(份)	5.87	36.01
云南生菜(份)	4.46	56.9	紫茄子(2)	6.01	23.21
苋菜	3.95	13.71	螺丝椒(份)	3.46	18.69
虫草花(份)	3.25	4.25	西兰花	12.12	22.23
云南油麦菜(份)	4.22	35.46	高瓜(1)	11.96	5.63
枝江青梗散花	5.78	4.92	西峡花菇(1)	24.93	8.05
小青菜(1)	4.87	10.26	七彩椒(2)	17.56	3.33
小皱皮(份)	2.37	22.53	-	-	-

八、 问题四的模型建立与求解

商超的补货和定价决策受到多种因素的影响。从供应商角度来说，供应商的履约能力和产品质量、采购成本、运输成本、到货速度等都会对商超的补货速度和定价产生一定的影响。从商家角度来说，蔬菜的保质期、会员折扣、货架陈列方式、库存量、库存周转率等都会对蔬菜的销量产生较大的影响，进而影响商家的补货决策和定价点。从消费者角度来说，客流量、退货率、满意程度都会直接或间接对蔬菜的销量产生影响。此外，气温、降水量、节假日、政府消费券、竞争对手的蔬菜价格、市场销售价格也会影响商超蔬菜的销量，也是商家在制定补货和定价决策时需要考虑的问题。

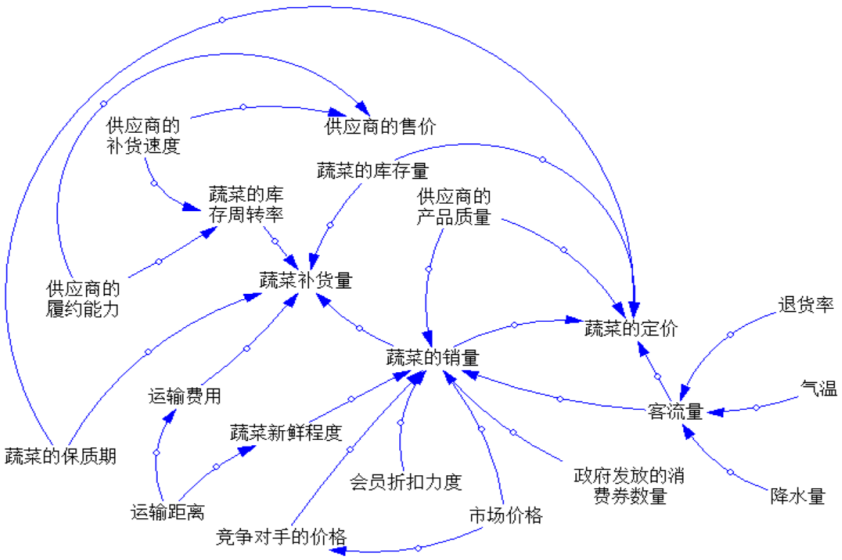


图 7 商家定价和补货决策的因果回路图

8.1 蔬菜的库存量

蔬菜不同于商超其他产品，蔬菜类商品具有难储存、易腐坏的特性，还极易受外界因素例如温度、湿度等的影响。恰当的蔬菜库存数量，能够减少蔬菜缺货而带来的损失，避免影响顾客的消费体验；能够防止蔬菜产品积压，增加仓储成本，避免由于蔬菜品质降低而造成的滞销。适量的库存还能够保证商超供货能力的稳定，降低极端天气等不可控因素造成的供应商无法及时供货带来的风险。因此，了解商超的库存量，可以帮助商家了解蔬菜商品的库存情况，从而制定更加合理的补货策略，寻找出更加准确的定价点。

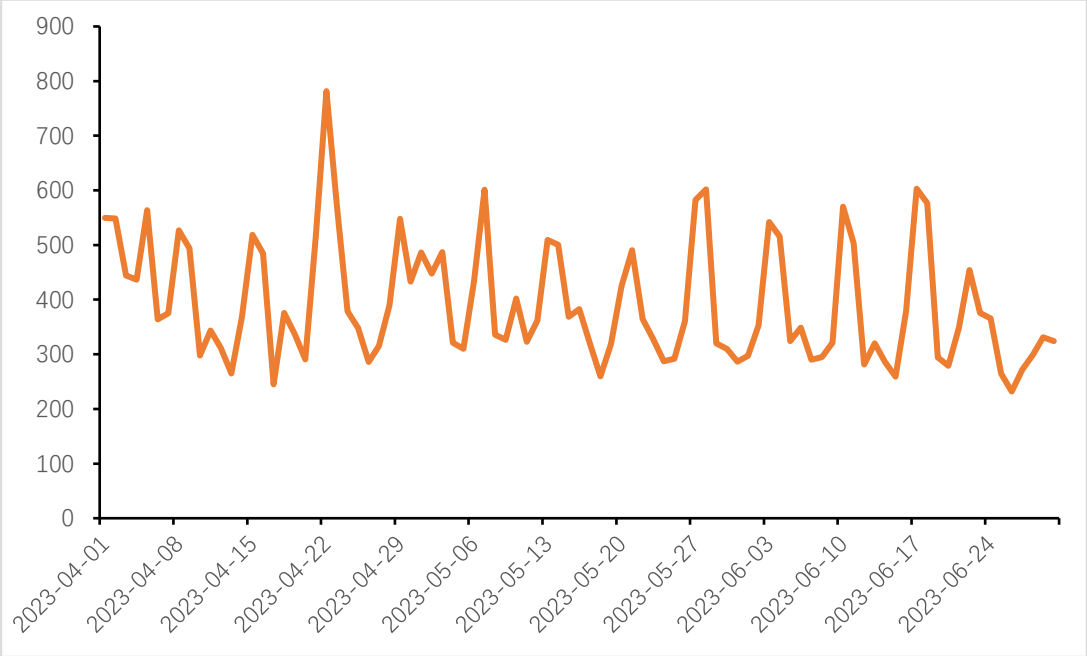


图 8 商超蔬菜的总销量

商超蔬菜的销量在不同时间段呈现间断式或者跳跃式的变化，顾客对于蔬菜的需求属于季节性需求类型。对于商超蔬菜库存量数据可以采用定性和定量两种预测方法。在缺少以往商超的蔬菜库存数据的情况下，可以通过定性分析，根据商超蔬菜区销售

人员的经验对蔬菜的库存进行一个大概的估计。在掌握部分以往蔬菜库存数据的情况下，应用时间序列分析法、线性回归分析法等分析方法对蔬菜库存进行预测。

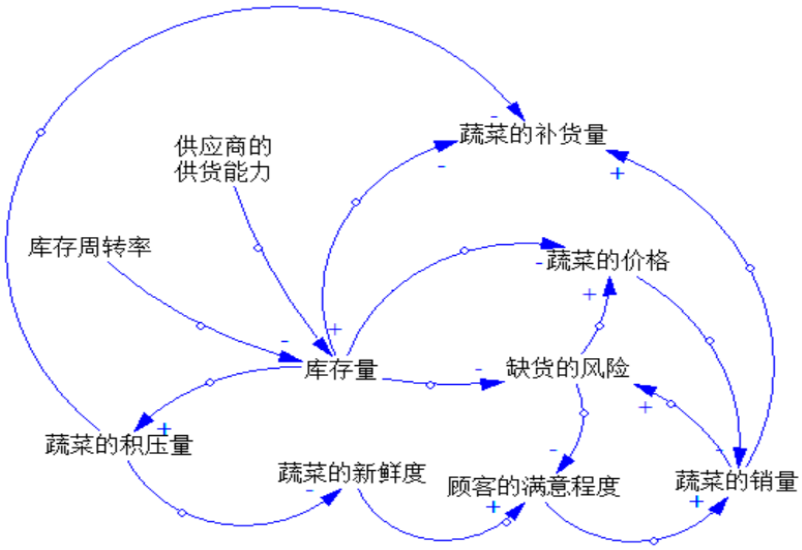


图 9 库存量对补货量和定价的因果回路图

蔬菜库存量可以明确商家在销售关系中的供需关系，库存量与销售量的结合能够减少模型中数据所表示的含义的模糊性，使得商超蔬菜的补货策略更加合理。清晰的供需关系能够更好地反映出蔬菜合适的价格，为商家做出定价策略提供更合理的依据。

8.2 每日的天气状况

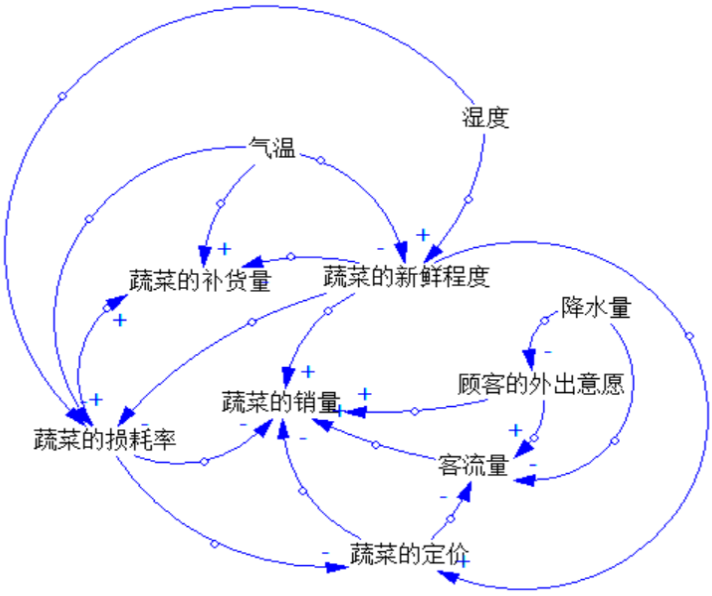


图 10 天气对补货量和定价的因果回路图

每日的天气状况好坏关系到商超蔬菜销量的多少，降水量、空气湿度、气温等数据也会对商家的补货和定价策略产生影响。目前的模型中只能了解到商品的损耗率对于蔬菜定价产生影响，实际上，降水量的增加会降低顾客的出行意愿，减少商超的客流量，干燥的空气降低蔬菜的新鲜程度，高温会加速蔬菜的腐烂，从而影响商超对蔬

菜的定价和蔬菜的损耗率。本模型中未考虑到这些因素对于蔬菜销量的影响，使得部分结果存在局限性。天气状况数据的记录可以使蔬菜销量的影响因素更加明确，更好的反映出蔬菜的销售情况和恰当的补货点，帮助商家制定更加合理的售价。

九、 模型的分析与检验

9.1 针对问题一模型的检验：

在处理问题一之前，首先对附件表格 1、2 中的数据进行了预处理，对于缺失值用 0 进行插值。同时选取了 Spearman 相关系数而非 Pearson 相关系数排除了不同蔬菜品类之间的重量差异对于相关性的影响，分析方法相对于传统的相关性分析更加合理。同时分析结论可以与现实相联系，进一步证明了分析的合理性。

9.2 针对问题二模型的检验：

针对问题二，首先使用 OLS 回归分析销售总量和成本加成定价之间的关系，并且通过 t-检验和 F-检验的显著性证明了二者间关系并不显著的结论，使得分析结果更显真实有效。同时使用 Prophet 算法时，多次调整不同参数选择变点的数量和范围最后得到了合适的参数值。

9.3 针对问题三模型的检验：

在第二问预测的基础上，在规划问题中增加了最小商品重量以及单品数量控制两个限制条件，为了使得 Prophet 算法更加贴合真实值，重新调整了参数，与 ARIMA 方法和线性预测的方法得到的结果相比较更加合理。

9.4 针对问题四模型的检验：

在问题四中，从两个角度出发，分别从蔬菜的库存量和每日的天气情况出发，以不同的逻辑说明收集这两种信息对于制定补货和定价决策的帮助，显得模型更加全面。

十、 模型的评价、改进与推广

10.1 模型的优点：

1.在数据预处理阶段，使用 0 对缺失值进行插值，并且使用归一化对原始数据进行处理。

2.在问题一中使用相关系数热力图，将不同变量之间的相关性变得更加显而易见，并且使用了 Spearman 相关系数，防止不同种类的蔬菜间由于重量差异较大而产生的相关性分析误差。

3.使用系统聚类法对各蔬菜品类中各单品进行聚类分析，考察了单品间的相关性，并且对于一些现象做出了基于现实的合理推断。

4.在使用 FB-prophet 方法进行预测时，不论是对品类还是单品进行预测，都对参数进行了敏感性分析，选择了最优参数。

5.在进行规划运算时，全面考虑了问题中的所有限制条件，使得模型最终求解得到的答案准确性较高。

10.2 模型的缺点：

在问题二、问题三中，预测售价及批发价的时间序列仅有近三个月，若使用更加长期的数据进行预测，可能 FB-Prophet 算法预测的效果会有进一步改善。

10.3 模型的展望：

对于问题二、问题三，可以尝试使用其他时间序列预测算法如 LSTM 算法、ARIMA 算法等，得到不同的预测结果和 FB-Prophet 的结果做比较，观察哪种时间序列算法效果更好。

十一、 参考文献

[1] Jha, B. K., & Pande, S. (2021, April). Time series forecasting model for supermarket sales using FB-prophet. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 547-554). IEEE.

[2] Saiktishna, C., Sumanth, N. S. V., Rao, M. M. S., & Thangakumar, J. (2022, May). Historical Analysis and Time Series Forecasting of Stock Market using FB Prophet. In 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1846-1851). IEEE.

[3] Baumol, W. J., & Bradford, D. F. (2005). Optimal departures from marginal cost pricing. In Transport Economics (pp. 194-217). Routledge.

[4] 原云霄,王宝海,宋洁.气温变动对我国蔬菜价格指数影响的实证分析——基于全国 173 个城市最低气温与 28 种重点监测蔬菜批发均价[J].数学的实践与认识,2019,49(14):263-269.

[5] 翟志宏,江民星,常春英.降水对蔬菜价格的冲击效应——以广州为例[J].资源科学,2021,43(02):304-315.

[6] 张瑞龙,杨肖丽.消费券影响蔬菜批发价格吗——来自中国 50 个批发市场的证据[J/OL].农业技术经济:1-19[2023-09-09].https://doi.org/10.13246/j.cnki.jae.20220420.001.

附录

附录 1

介绍：支撑材料的文件列表

Prophet 算法图片.rar :R 语言中使用 Prophet 算法预测售价和批发价时生成的时间序列图形。

Python 代码.rar:建模过程中使用的 python 代码

linear.py：作出各个品类/单品销量比例按价格的线性分布图像以及返回回归方程；

main.py：导入了所有用到的模块，是代码即写即运行的文件；

stored.py：存储所有运行过的代码，保留备份，以备同类需求直接借鉴、同时保留了 data.dat 文件中的内容的生成代码，提供了其中变量的列表和数据结构。

R 代码.rar: 建模过程中使用的 R 代码

预测型 R 代码:使用 Prophet 算法对售价和批发价进行预测的文件；

聚类型 R 代码:使用系统聚类法对不同蔬菜品类中的单品进行聚类的文件；

箱线图 R 代码:绘制箱线图的文件；

相关性分析 R 代码:进行绘制相关系数矩阵,相关系数热力图等相关性分析的代码。

SPSS 结果.rar:

描述性统计数据.sav:使用 SPSS 对数据进行描述性分析的文件；

蔬菜类商品相关系数矩阵及其显著性分析.sav:使用 SPSS 输出相关系数矩阵及显著性的文件。

品类和单品预测数据.rar:

大类月度销售数据(归一化后).xlsx:对原始月度销售数据进行归一化后的数据；

预测类表格:使用 Prophet 算法进行预测得到预测数据的表格。

附录 2

问题 2 和问题 3 的代码，使用 Python 语言编写

```
import shelve
import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import prophet
import plotly.express as px
import openpyxl
from openpyxl.utils import get_column_letter as column
import pylab as plb

shelf=shelve.open('data')



global data,infodic
data=pd.read_excel('for_analysis.xlsx',sheet_name='Sheet4')
```

```
    infodic=dict()

    targetlst=[
        '茄类',
        '水生根茎类',
        '花叶类',
        '花菜类',
        '辣椒类',
        '食用菌'
    ]

    def process(target:str)->list():
        # 1.数据输入
        x = np.array(data[target+'ap'])
        y = np.array(data[target+'ar'])
        # 2.返回值为各项系数
        first_step=np.polyfit(x,y,1)
        # 3.获得函数表达式
        p1=np.poly1d(first_step)
        # 4.代入函数值
        y_new=p1(x)

infodic[target]='proportion=('+str(first_step[0])+'*x+'+str(first_step[1])+')/100'
        # 5.可视化绘图
        plb.title(infodic[target],color='r')
        plb.xlabel('price',color='r')
        plb.ylabel('prop*100',color='r')
        fig1=plb.plot(x,y,'*',label='OLD FIGURE')
        fig2=plb.plot(x,y_new,'r',label='FITED FIGURE')
        plb.legend(loc=3, borderaxespad=0., bbox_to_anchor=(0, 0.873))
        plt.show()
        plt.close()
        return first_step

    for target in targetlst:
        process(target)

    data=pd.read_excel('for_analysis.xlsx',sheet_name='Sheet6')
    process(targetlst[1])

    shelf['class_linear_1']=infodic

    for i in shelf['class_linear_1']:
        print(i,shelf['class_linear_1'][i])
```

```
    shelf.close()
```

附录 3

问题 1 代码，使用 Python 语言 编写

```
import openpyxl
from openpyxl.utils import get_column_letter as column
import shelve
import datetime
import prophet

data=shelve.open('data')

table4=data['tabele4']

def getInfo()->dict:
    infoDic=dict()
    infosheet=openpyxl.load_workbook('附件 1.xlsx')['Sheet1']
    row=2
    code=infosheet['A'+str(row)].value
    name=infosheet['B'+str(row)].value
    classcode=infosheet['C'+str(row)].value
    classname=infosheet['D'+str(row)].value
    while code:
        infoDic[code]=name
        infoDic[classcode]=infoDic.get(classcode,[classname])
        infoDic[classcode].append(code)
        row+=1
        code=infosheet['A'+str(row)].value
        name=infosheet['B'+str(row)].value
        classcode=infosheet['C'+str(row)].value
        classname=infosheet['D'+str(row)].value
    return infoDic
# data['codeDic']=getInfo()

def proFj2()->list:
    table=list()
    sheet=openpyxl.load_workbook('附件 2.xlsx')['Sheet1']
    row=2
    date=sheet['A'+str(row)].value    # datetime.datetime
    time=sheet['B'+str(row)].value    # str
    code=sheet['C'+str(row)].value    # str
```

```
        amount=sheet['D'+str(row)].value        # float
        price=sheet['E'+str(row)].value # float
        type=sheet['F'+str(row)].value    # str
        isdiscount=sheet['G'+str(row)].value        # str
        while date:

table.append([[date.year,date.month,date.day],time,code,amount,price,type,isdiscount])
                row+=1
                date=sheet['A'+str(row)].value    # datetime.datetime
                time=sheet['B'+str(row)].value    # str
                code=sheet['C'+str(row)].value    # str
                amount=sheet['D'+str(row)].value        # float
                price=sheet['E'+str(row)].value # float
                type=sheet['F'+str(row)].value    # str
                isdiscount=sheet['G'+str(row)].value        # str
        return table
# data['table']=proFj2()



def proFj3()->list:
        table=list()
        sheet=openpyxl.load_workbook('附件 3.xlsx')['Sheet1']
        row=2
        date=sheet['A'+str(row)].value    # datetime.datetime
        code=sheet['B'+str(row)].value    # str
        price=sheet['C'+str(row)].value # float
        while date:
                table.append([[date.year,date.month,date.day],code,price])
                row+=1
                date=sheet['A'+str(row)].value    # datetime.datetime
                code=sheet['B'+str(row)].value    # str
                price=sheet['C'+str(row)].value # float
        return table


# data['Q1_dic']=Q1_dic
# data['Q1_dic_class']=Q1_dic_class

codeDic=dict(data['codeDic'])
table=list(data['table'])
Q1_dic=dict()
Q1_dic_class=dict()
for line_num in range(len(table)):
```

```
        line=table[line_num]
        code=line[2]
        month=str(line[0][0])+'年'+str(line[0][1])+'月'
        amount=line[3]

Q1_dic[(code,codeDic[code],month)]=Q1_dic.get((code,codeDic[code],month),0)+amou
nt
        for i in codeDic.keys():
            if code in codeDic[i]:

Q1_dic_class[(i,codeDic[i][0],month)]=Q1_dic_class.get((i,codeDic[i][0],month),0)+am
ount
                break
            else:
                continue
# data['Q1_dic']=Q1_dic
# data['Q1_dic_class']=Q1_dic_class



# data['Q1_dic_hour']=Q1_dic_hour
# data['Q1_dic_class_hour']=Q1_dic_class_hour

codeDic=dict(data['codeDic'])
table=list(data['table'])
Q1_dic_hour=dict()
Q1_dic_class_hour=dict()
for line_num in range(len(table)):
        line=table[line_num]
        code=line[2]
        hour=int(line[1].split(':')[0])
        amount=line[3]

Q1_dic_hour[(code,codeDic[code],hour)]=Q1_dic_hour.get((code,codeDic[code],hour),0
)+amount
        for i in codeDic.keys():
            if code in codeDic[i]:

Q1_dic_class_hour[(i,codeDic[i][0],hour)]=Q1_dic_class_hour.get((i,codeDic[i][0],hour)
,0)+amount
                break
            else:
                continue
# data['Q1_dic_hour']=Q1_dic_hour
# data['Q1_dic_class_hour']=Q1_dic_class_hour
```

```
# data['Q1_class_type_times']=Q1_class_type_times
# data['Q1_class_discount_times']=Q1_class_discount_times

codeDic=dict(data['codeDic'])
table=list(data['table'])
Q1_class_type_times=dict()
Q1_class_discount_times=dict()
for line_num in range(len(table)):
	line=table[line_num]
	code=line[2]
	month=str(line[0][0])+'年'+str(line[0][1])+'月'
	amount=line[3]
	type=line[5]
	isdiscount=line[6]
	#
Q1_dic[(code,codeDic[code],month)]=Q1_dic.get((code,codeDic[code],month),0)+amou
nt
	for i in codeDic.keys():
		if code in codeDic[i]:

Q1_class_type_times[(i,codeDic[i][0],month,type)]=Q1_class_type_times.get((i,codeDic
[i][0],month,type),0)+1

Q1_class_discount_times[(i,codeDic[i][0],month,isdiscount)]=Q1_class_discount_times.
get((i,codeDic[i][0],month,isdiscount),0)+1
				break
		else:
				continue
# data['Q1_class_type_times']=Q1_class_type_times
# data['Q1_class_discount_times']=Q1_class_discount_times



Q1_class_type=data['Q1_class_type']
Q1_class_discount=data['Q1_class_discount']

rows=dict()
row_count=2
wb=openpyxl.Workbook()
sheet=wb['Sheet']
for key in Q1_class_discount_times.keys():
```

```
        code=key[0]
        name=key[1]
        month=key[2]
        isdiscount=key[3]
        times=Q1_class_discount_times[key]
        try:
            rows[(code,isdiscount)]
        except:
            rows[(code,isdiscount)]=rows.get((code,isdiscount),row_count);row_count+=1
        row=str(rows[(code,isdiscount)])
        col=column((int(month[:4])-2020)*12+int(month[5:-1])-3)
        sheet['A'+row]=code
        sheet['B'+row]=name
        sheet['C'+row]=isdiscount
        sheet[col+'1']=month
        sheet[col+row]=times
wb.save('品类_按月_是否折扣_次数.xlsx')
wb.close()


# dic_date_info
table2=data['table']
table3=data['table3']
dic2=dict()
dic3=dict()
dic_date_info=dict()

for info3 in table3:
        date=tuple(info3[0])
        code=info3[1]
        price=info3[2]
        dic3[(date,code)]=price
for info2 in table2:
        date=tuple(info2[0])
        # time=info2[1]
        code=info2[2]
        amount=info2[3]
        price=info2[4]
        # type=info2[5]
        # isdiscount=info2[6]
        infolst=dic2.get((date,code),[{}])
        infolst[0][price]=infolst[0].get(price,0)+amount
        dic2[(date,code)]=infolst.copy()
for key in dic2.keys():
```

```
        infolst=dic2[key]
        try:
            infolst.append(dic3[key])
        except:
            infolst.append(dic3.get(key,0))
            print(key)
        dic_date_info[key]=infolst


# dic_class_date
classlst=[]
for key in codeDic.keys():
        try:
            codeDic[codeDic[key][1]]
            classlst.append(key)
        except:
            continue
print(classlst)

dic_class_date=dict()
for key in dic_date_info.keys():
        date=key[0]
        code=key[1]
        amount=sum(list(dic_date_info[key][0].values()))
        featureValue=0
        for price in dic_date_info[key][0].keys():
            featureValue+=price*dic_date_info[key][0][price]
        for classcode in classlst:
            if code in codeDic[classcode]:
                infolst=dic_class_date.get((date,classcode),[0,0,0])

amountclass=infolst[0];featureValueclass=infolst[1];averageprice=infolst[2]

amountclass+=amount;featureValueclass+=featureValue;averageprice=featureValueclass/
amountclass

dic_class_date[(date,classcode)]=[amountclass,featureValueclass,averageprice]      # 麻
了这里
            else:
                continue



def write(wb:openpyxl.Workbook,target:tuple):
```

```
	row_num=2
	rows=dict()
	wb.create_sheet('单品_'+str(target))
	sheet=wb['单品_'+str(target)]
	for key in dic_date_info.keys():
		date=key[0]
		if date[:2]==target:
			code=key[1]
			name=codeDic[code]
			pricedic=dic_date_info[key][0]
			try:
				row=rows[code]
			except:
				rows[code]=row_num;row_num+=1
			row=str(rows[code])
			col=column(date[2]+2)
			# for i in pricedic.keys():
			#	amount+=pricedic[i]
			# price_average=price_average/sum(list(pricedic.values()))
			sheet[col+row]=sum(list(pricedic.values()))
			sheet['A'+row]=code
			sheet['B'+row]=name
			sheet[col+'1']=str(date[0])+'年'+str(date[1])+'月'+str(date[2])+'日'
		else:
			continue
	return wb

targetlst=[
	(2020,7),
	(2021,7),
	(2022,7),
	(2023,4),
	(2023,6),
	(2023,6)
]

# class_batch_rate
class_batch_rate=dict() # only latest 7 days are calculated
for key in dic_date_info.keys():
	date=key[0]
	code=key[1]
	amount=sum(list(dic_date_info[key][0].values()))
	featurevalue=amount*table4[code]
	for classcode in classlst:
```

```
            if date in targetlst:
                pass
            else:
                continue
            if code in codeDic[classcode]:
                infolst=class_batch_rate.get((date,classcode),[0,0,0])
                amountclass=infolst[0];featurevalueclass=infolst[1]

amountclass+=amount;featurevalueclass+=featurevalue;rate=featurevalueclass/amountcl
ass
                class_batch_rate[(date,classcode)]=[amountclass,featurevalueclass,rate]
            else:
                continue
data['class_batch_rate']=class_batch_rate

data.close()
```

附录 4

使用 Python 语言编写，工作台。

```
import openpyxl
from openpyxl.utils import get_column_letter as column
import shelve
import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import prophet

data=shelve.open('data')

if __name__=="__main__":
    # do
    pass

data.close()
```

附录 5

使用 R 语言编程，prophet 预测代码，以白玉菇（袋）为例，其余见支撑材料

```
library(prophet)
library(readxl)
df<-read_xlsx('白玉菇（袋）批发价.xlsx')
m<-prophet(df,changepoint.prior.scale = 0.8)
future<- make_future_dataframe(m,periods=7)
forecast<- predict(m,future)
plot(m,forecast)
print(forecast[c('ds','yhat','yhat_lower','yhat_upper')])
```

附录 6

使用 R 语言编程，聚类代码，以茄类为例，其余见支撑材料

```
    library(cluster)# 加载包
    library(readxl)
    library(rattle)
    library(factoextra)
    data <- read_excel("茄类商品原始表格.xlsx")
    x=as.data.frame(data)
    rownames(x)=c("紫茄子(2)", "青茄子(1)", "紫圆茄", "大龙茄子", "花茄子", "长线茄", "青茄子(2)", "紫茄子(1)", "圆茄子(1)", "圆茄子(2)")
    df=scale(data[-1])
    d=dist(df)
    print(d)
    hc2=hclust(d,"complete")#最长距离法，默认的联接方法
    plot(hc2,hang=-1,rotate=T)#最长距离法的树状图
```

附录 7

相关性分析代码，使用 R 语言编程

```
library(corrplot)
library(psych)
library(readxl)
library(Hmisc)
data <- read_excel("相关性分析原始表格.xlsx")
mycor=cor(data,method ="spearman") #计算相关系数
mycor
round(mycor,2)  # 输出相关系数，保留两位小数
```

```
plot(data[,-c(1)]) #变量太多，可视化效果很糟糕
corrplot(mycor,tl.col="black",type="upper")  #相关系数可视化，需要调用 corrplot
```

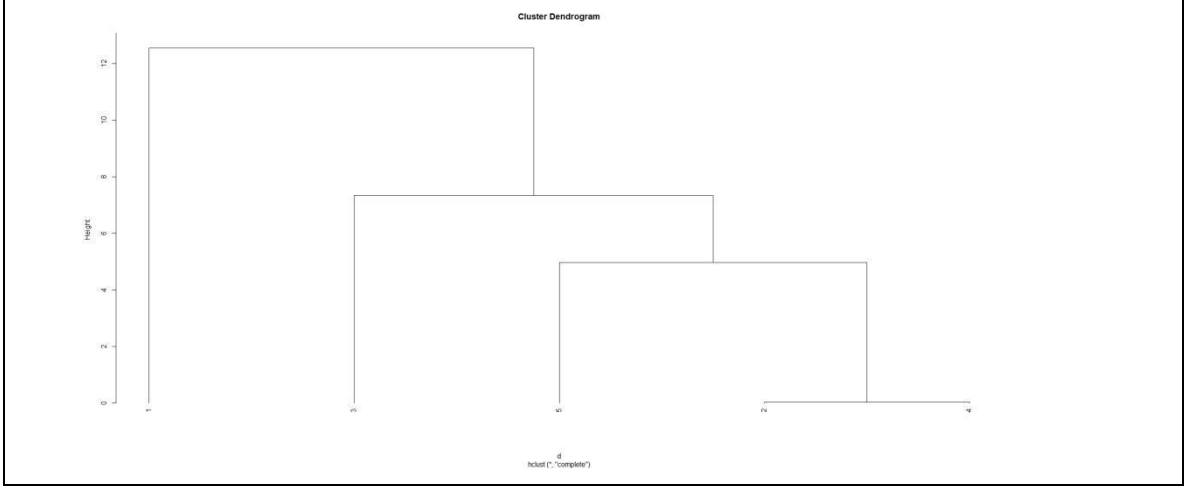
附录 8

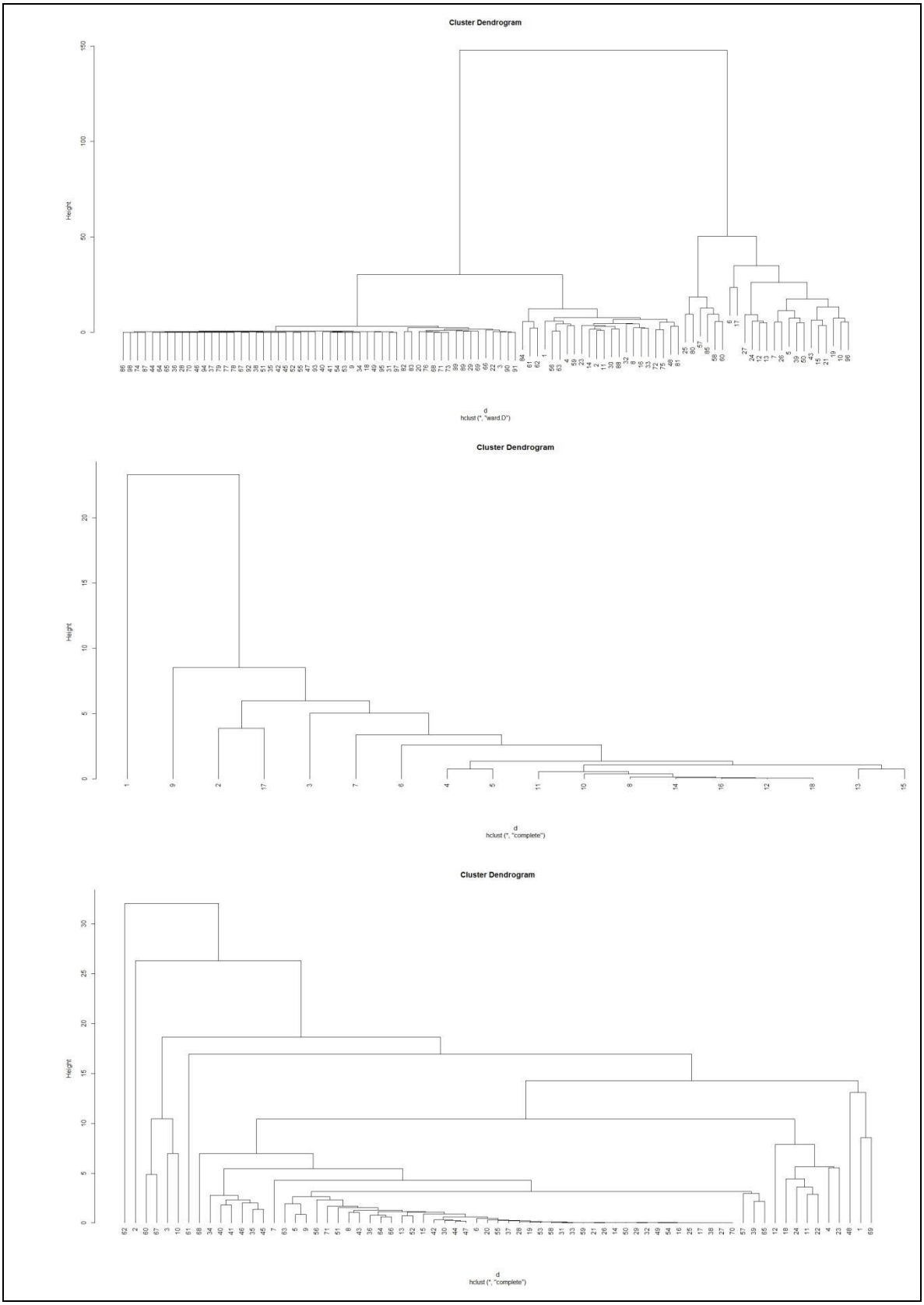
箱线图绘制代码，使用 R 语言编程

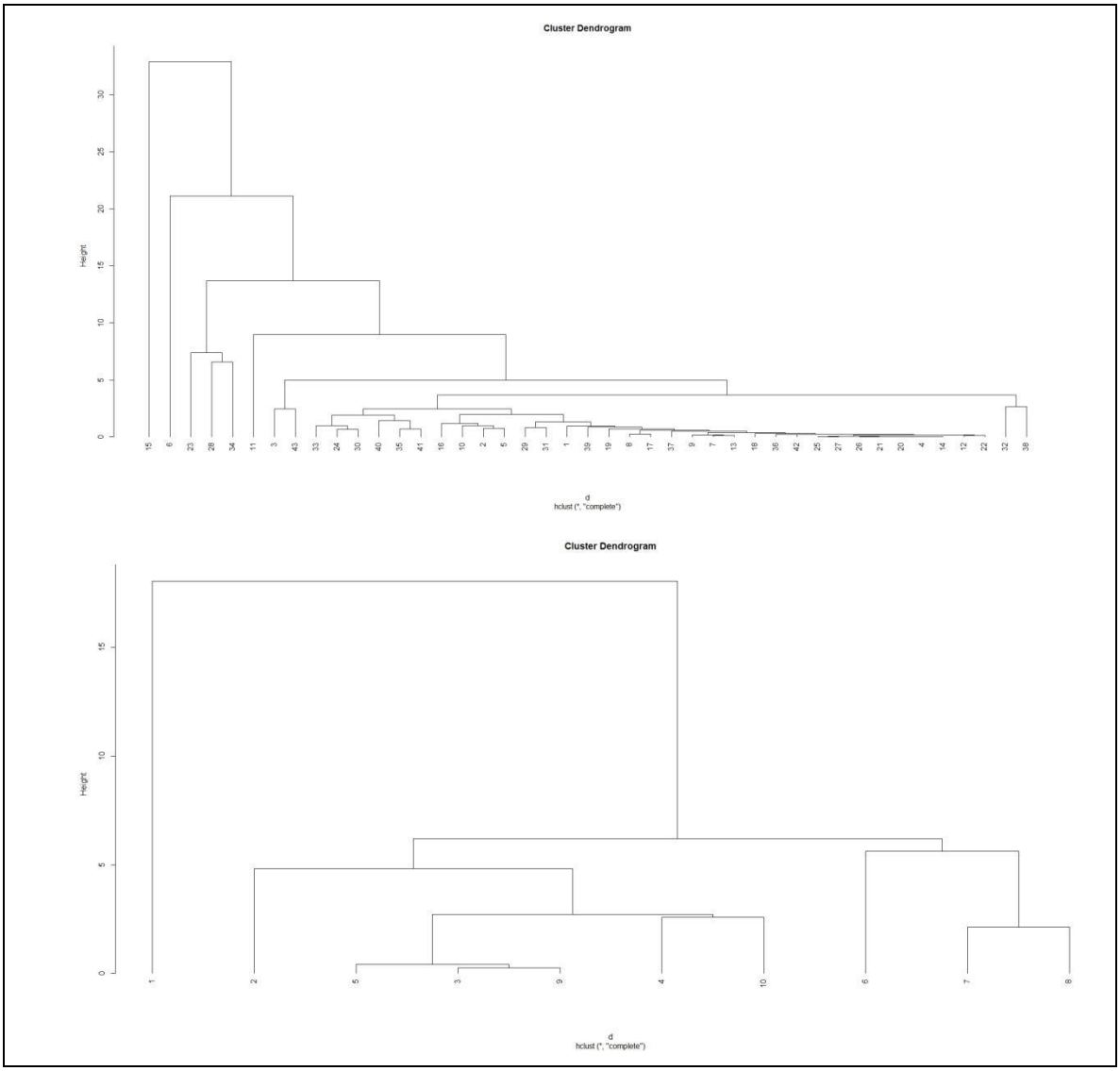
```
library(ggplot2)
library(readxl)
data<-read_excel("蔬菜类商品分类箱线图原始表格.xlsx")
library(fBasics)
attach(mtcars) #  固定数据集
data$种类=as.factor(data$种类)
p = ggplot(data, aes(x=种类, y=销量,color=种类)) + geom_boxplot(outlier.colour="red", outlier.shape=18,outlier.size=4)
p
p + geom_jitter(shape=16, position = position_jitter(0.2))
```

附录 9

蔬菜品类中单品聚类结果




Cluster Dendrogram
Height
d
hclust (*, "ward.D")
Cluster Dendrogram
Height
d
hclust (*, "complete")
Cluster Dendrogram
Height
d
hclust (*, "complete")


Cluster Dendrogram
Height
0
5
10
15
20
25
30
15 6 23 28 34 11 3 43 33 24 30 40 35 41 16 10 2 5 29 31 1 39 19 8 17 37 9 7 13 18 36 42 25 27 26 21 20 4 14 12 22 32 38
d
hclust (*, "complete")
Cluster Dendrogram
Height
0
5
10
15
1 2 5 3 9 4 10 6 7 8
d
hclust (*, "complete")