2025年（第一届）数智杯数据分析大赛

**零售门店销售数据深度挖掘与可视化分析**

|  |  |
| --- | --- |
| 参赛学校： | 广西师范大学 |
| 论文题目： | 零售门店销售数据深度挖掘与可视化分析 |
| 参赛队员： | 安奇 陈思铭 任峥 |
| 指导老师： | 高翔宇 |

**零售门店销售数据深度挖掘与可视化分析**

**摘 要**

**针对问题一**，本文以浙江省万客缘佳乐福店**2021.1.1-2022.12.8**的销售数据为样本，通过帕累托分析识别关键品类与促销活动，为零售企业精细化运营提供理论依据。首先，我们建立了四级**品类管理体系**（一级：医疗保健/生鲜/非食/食品；二级：16个子类如冲调食品、烟酒水饮等），通过程序对每个品类的销售额和销售数量进行精确统计，并将计算结果进行**可视化分析**。数据分析显示，在四大一级品类中，食品品类贡献了66.73%的总销售额，显著高于其他品类；进一步细分发现，烟酒水饮（26.46%）、冲调食品（4.24%）、粮油调味（18.03%）、休闲食品（11.56%）、家居用品（7.59%）、冷藏乳制品（5.44%）、卫生用品（6.50%）和个人清洁（5.09%）七大品类合计贡献约80%的销售额，符合**帕累托法则**。在商品层面，通过排序分析查找出各个关键品类下的**灯塔商品**，其中，剑光黄酒、蒙牛鲜牛奶和某品牌乳酸菌饮品位列前三。针对促销效果，我们运用时间序列分析法，追踪了爆款商品在促销期间（标注了重要节假日的真实日期）的价格波动与销量变化，并且构建了**需求-价格弹性模型，**精准量化价格变动对市场需求的影响。实证数据显示：酱油调味料在促销前后相同时间内，促销后的销售额为促销前销售额的3.857倍，较促销前增长了285.7%，由此可知，部分促销活动对销售业绩有显著的影响。

**针对问题二**，首先，通过业务理解选取了碳酸饮料、冷藏水饮、肉类具有典型季节性的品类（依据商品属性与消费常识判断）。使用Python的pandas库从原始数据中提取这些品类的每日销售记录，并按月份**聚合计算**销售额均值。为直观展示**季节性规律**，并绘制了各品类12个月的销售曲线图，标注了年度峰值（如碳酸饮料在2021年7月达到峰值18129.33元）和谷值（肉类在2021年12月降至年度最低1615.6元）。通过时间序列分解，发现碳酸饮料的季节性指数夏季（6-8月）平均比冬季高58%，而肉类则呈现相反的-32%差异。为验证模型预测能力，我们构建了**SARIMA**（**季节性自回归积分滑动平均模型**）模型（参数设置：d=1,D=1,m=12,p=2,P=1,q=1,Q=1），以整年年数据训练模型预测2025年销售，证明模型有效性后，最终用于预测2025年6月至2026年6月的销售走势，为库存决策提供数据支持。

**针对问题三**，本研究采用关联规则分析中的**Apriori算法**，对某零售门店约100万条交易记录进行数据挖掘。首先，通过计算支持度以识别高频出现的商品品类与组合；在此基础上，进一步计算置信度以量化品类间的关联强度，从而发现具有显著性的购买依存关系。在设置支持度阈值为0.3的前提下，“家居用品”为核心品类（支持度0.5278）。关联性最强的商品组合为“家居用品→烟酒水饮”，其置信度高达25%，即购买家居用品的顾客中有四分之一会同时购买烟酒水饮；其次为“家居用品→粮油调味”，置信度为22%；“休闲食品→家居用品”的组合也表现出15%的显著关联性。

**针对问题四**，我们需要采集一些新的数据，引用一些新的技术，但是最终应该为零售店实现智能自动化决策服务。销售的本质是服务于客户，我们可以引入动态数据库，搭建本地或**云端数据库**平台，并且引入相关预测模型来预测未来一段时间SKU级别的销量，实现自动补货，实时调整营销策略等方案。

**关键词** 帕累托法则 需求-价格弹性模型 季节性自回归积分滑动平均模型

**一 问题背景和重述**

**1.1问题背景**

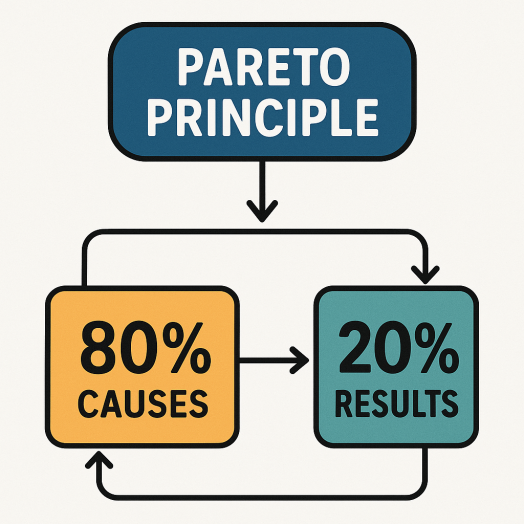
在当今高度竞争的零售市场中，零售门店面临着商品同质化严重、消费者需求多样化、市场竞争激烈等多重挑战。如何通过数据驱动的营销策略提升门店运营效率、优化资源配置并增强市场竞争力，成为零售企业生存发展的关键问题。本文结合实际零售门店的业务场景，期望能够为零售门店的营销策略优化提供有价值的参考。

**1.2问题重述**

在当今竞争激烈的零售市场中，零售门店面临着诸多挑战与机遇。如何在海量的商品和复杂的市场环境中制定有效的营销策略，以提升销售额、优化库存管理并增强顾客满意度，是每个零售企业亟待解决的问题。本次竞赛旨在通过对某零售门店多年销售数据的深入分析，探索出科学合理的营销策略，助力门店在激烈的市场竞争中脱颖而出。

该零售门店作为一家综合性零售企业，拥有丰富多样的商品品类，涵盖了生鲜、食品、日用品等多个领域，满足了不同消费者群体的多样化需求。门店的运营时间较长，积累了大量的销售数据，这些数据涵盖了从 2021 年 1 月 1 日至 2025 年 4 月 30 日的详细销售记录，包括时间维度、企业信息、门店信息、订单信息、商品基础信息及商品销售信息等多维度字段，完整呈现了门店 - 订单 - SKU 级别的商品销售流水数据。这些数据为深入分析门店的销售情况、消费者行为模式以及营销活动效果提供了坚实的基础。

**问题1** 零售门店中通常存在帕累托法则，即约20%的商品贡献了80%的销售额，像生鲜、乳制品、畅销零食以及新品类往往占据门店的主要销售额；同时，20%的促销活动能够带动80%的销量增长，例如节假日促销、爆品折扣等关键营销活动对业绩有着显著影响。基于此，请尝试找出该门店的关键品类或商品，以及有效的促销活动，并进行深入分析。



**图1.1 帕累托法则**

**问题2** 零售门店的商品或品类消费往往具有一定的季节性特征。例如，碳酸饮料在炎热的7、8月份消费量较高，而过高的气温则可能导致整体肉类消费量降低。请分析各关键品类或关键单品的分布规律，以更好地把握季节性因素等关键因素对销售的影响，同时预测关键品类或单品在下一年度即2025年6月至2026年6月的销售情况。

**问题3** 在门店商品销售过程中，不同品牌类或不同单品之间，甚至与关键事件之间可能存在一定的关联关系。例如，曾经有“啤酒 + 尿不湿”的经典案例，以及在飓风来临前蛋挞与应急用品的搭配销售，这些关联关系对优化商品陈列及开展联合促销具有积极意义。请分析不同品类或不同单品之间的关联规则，并结合问题1和问题2的分析结果，给出下一年度即2025年6月至2026年6月该门店的营销策略建议。

**问题4** 在海量数据积累与人工智能技术飞速发展的当下，为了助力门店营销策略的科学化与智能化升级，除了现有的数据资源和营销手段之外，您认为该门店还应采集哪些关键数据，并引入哪些前沿技术手段，以推动营销决策向智能自动化方向迈进？请给出你的见解及相应理由。

**二 问题分析**

**2.1 问题一分析**

对于问题一，我们对问题中的帕累托法则进行了解读，其又被称为“80/20”法则，在许多情况下，约80%的结果（产出、收益或影响）往往由20%的原因（投入、资源或因素）所驱动。我们首先将数据集中的商品进行了品类分级，通过系统化的分类管理，优化资源分配。我们使用程序求解出各个品类的销售额和销售数量，并将结果保存至相应的文件中，直观地得到门店的关键品类和关键商品。其中，一级品类中（医疗保健、生鲜、非食、食品），食品品类成为门店销售额增长的核心驱动力，为销售额增长做出了卓越贡献；在二级品类中，冲调食品、烟酒水饮、粮油调味为核心品类，其总销售额满足帕累托法则，为销售额增长做出显著贡献。在关键商品分析方面，我们提取了门店在限定时间范围内各个商品（共16195种商品）的售卖情况，其中，我们发现剑光黄酒、乳酸菌饮品、蒙牛鲜牛奶等商品的销售额较其它商品占显著优势。在促销活动的分析方面，我们对门店爆款商品的单价随时间的变化曲线进行了详细分析，并且综合考量了实际日历中节假日等特殊时间，发现部分促销活动对门店销售业绩的提升有显著影响。

**2.2 问题二分析**

对于问题二，为了探寻商品消费的季节性特征，我们首先选择了具有显著季节特征的商品品类作为对象，例如在炎热夏季，碳酸饮料、冷藏水饮等商品品类的销售额相较于冬季会有明显的增加；而肉类食品则会出现夏季销售额低于冬季的情况。选择完具有明显季节特征的商品品类之后，我们通过程序将目标品类的数据从总数据集中提取出来，得到目标数据。为了直观查看商品的季节性特征，我们对目标数据进行了可视化操作，我们将目标商品的销售额随时间变化的关系绘制成图标，标记峰值、谷值。为了得到时间序列数据，我们在时间轴上取均匀时间间隔，并对相应间隔内的商品销售数据进行记录，之后我们对时间序列数据进行分析、得到序列的趋势、季节性等关键性质。在模型选择方面，我们选择SARIMA时间序列预测模型来预测2025年6月至2026年6月的销售情况。

**2.3 问题三分析**

对于问题三，为了深入探究门店商品销售过程中不同品类或单品的潜在关联关系，本研究采用关联规则挖掘中的Apriori算法，对销售流水数据进行系统分析。该算法能够从大规模交易数据中高效识别频繁出现的商品组合，并生成具有统计显著性的关联规则，适用于揭示类似“啤酒与尿不湿”的典型跨品类购买模式。分析过程中，首先计算各商品及商品组合的支持度以筛选频繁项集，进而基于置信度与提升度等指标评估规则强度，有效区分具有实际意义的关联模式与随机共现现象。通过可视化展示高频关联组合的网络结构，本研究为零售门店优化商品陈列策略、设计联合促销方案以及精准预测需求提供了数据驱动的决策依据，从而提升交叉销售机会与整体经营效益。

**2.4 问题四分析**

根据题意，我们思考了两个层面，分别是我们应该采集哪些关键的数据，以及我们应该引入哪些关键手段，赋能旧数据最后达到实现智能自动化决策的目的。在采集数据方面，我们考虑可以引入外部环境数据，例如天气数据、宏观经济与社交数据，外部环境会显著影响消费需求。例如天气会增加冰饮数量，雨天会提升速食产求。在技术层面，由于交易数据是静态数据，所以我们考虑引入动态云端数据库，可以实时追踪零售门店的销售数据，门店可以根据动态数据来调整相关销售策略。

# **三 模型假设**

1.商品销售数据具有明显的季节性特征，且该特征在预测周期内保持稳定

2.促销活动对销售的影响可以通过历史数据进行量化评估

3.不同品类之间的销售波动相互独立，不考虑跨品类影响

4.门店运营策略和商品结构在预测期内保持相对稳定

**四 定义与符号说明**

|  |  |  |
| --- | --- | --- |
| 符号 | 说明 | 单位 |
|  | 一级品类下第i个品类的销售数量 | 件 |
|  | 一级品类下第i个子品类的订单数 | 单 |
|  | 单笔订单销售数 | 件 |
|  | 商品的销售单价 | 元 |
| Lv | 品类级别 |  |
|  | 每个子品类的销售额 | 元 |
|  | 促销敏感度 |  |
|  | 促销期间平均销售数量 | 件 |
|  | 日常平均销售数量 | 件 |
|  | 促销时间 |  |
|  | 促销期间销售数量 | 件 |
|  | 日常销售数量 | 件 |
|  | 节假日单价 | 元 |
|  | 平日单价 | 元 |
|  | 季节性指数 |  |
|  | 原始观测值 |  |
|  | 序列平均值 |  |
|  | 季节性强度指数 |  |
|  | 异界差分后的新值 |  |
|  | 第t个月的销售额 | 元 |
|  | 前一个月的销售额 | 元 |
|  | 季节差分后最终值 |  |
|  | 当前月的一阶差分值 |  |
|  | 去年同月一阶差分值 |  |
| d | 普通差分阶数 |  |
| D | 季节性差分阶数 |  |
| m | 季节周期 |  |
| p | AR项阶数 |  |
| q | MA项阶数 |  |
| P | 季节性AR项阶数 |  |
| Q | 季节性MA项阶数 |  |
|  | 项集A的支持度 |  |
|  | 包含项集A的交易数 | 单 |
|  | 交易总数 | 单 |
|  | 置信度 |  |

# 

**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 关键品类商品分析**

为识别该零售门店的关键品类，我们采用分层聚合与帕累托分析法。商品品类分为四级，我们自顶向下逐层分析其销售贡献。设某一级品类下有K个子品类，每个子品类对应若干订单。定义第i个一级品类的总销售数量为：​

（1）

其中：

为该品类下的总订单数；

SQ(i,j)为第i个品类下第j笔订单的销售数量。

通过计算，我们得到一级品类的销售数量如下：

# **表5.1 一级品类销售数量**

|  |  |  |
| --- | --- | --- |
| **i** | **一级品类** | **销售数量** |
| 1 | 医疗保健 | 11360 |
| 2 | 生鲜 | 102709 |
| 3 | 非食 | 3263451 |
| 4 | 食品 | 2012319 |

# 

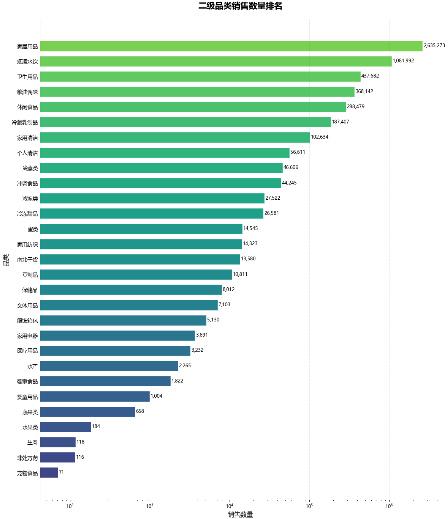
**图5.1 一级品类销售数量**

# 为进一步识别关键子品类，我们扩展至二级品类分析。定义第k个二级品类的销售数量为：

# （2）

# **表5.2 二级品类销售数量**

|  |  |  |
| --- | --- | --- |
| **i** | **二级品类** | **销售数量** |
| 1 | 个人清洁 | 56611 |
| 2 | 休闲食品 | 288479 |
| 3 | 保健品 | 8012 |
| 4 | 冲调食品 | 44245 |
| 5 | 冷冻甜品 | 26581 |
| 6 | 冷冻类 | 27522 |
| 7 | 冷藏乳制品 | 187407 |
| 8 | 冷藏类 | 46606 |
| 9 | 医疗用品 | 3232 |
| 10 | 南北干货 | 13580 |
| 11 | 卫生用品 | 437682 |
| 12 | 婴童用品 | 1004 |
| 13 | 婴童食品 | 1822 |
| 14 | 宠物食品 | 71 |
| 15 | 家居用品 | 2635273 |
| 16 | 家用清洁 | 102634 |
| 17 | 家用电器 | 3691 |
| 18 | 家用纺织 | 14323 |
| 19 | 文体用品 | 7103 |
| 20 | 服饰箱包 | 5130 |
| 21 | 水产 | 2265 |
| 22 | 水果类 | 184 |
| 23 | 烟酒水饮 | 1081992 |
| 24 | 生肉 | 118 |
| 25 | 粮油调味 | 368142 |
| 26 | 蔬果类 | 658 |
| 27 | 蛋类 | 14545 |
| 28 | 豆制品 | 10811 |
| 29 | 非处方药 | 116 |



**图5.2 二级品类销售数量**

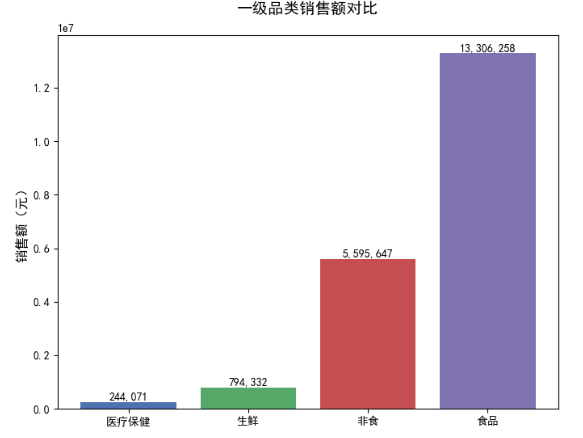
我们求出了各个品类的销售数量，再结合商品的单价，便可以求出各级品类下各子品类的销售额以及各品类占总销售额的百分比，设p为商品的销售单价（单位：元）,S(i)为每个子品类的销售额：

**表5.3 一级品类销售额**

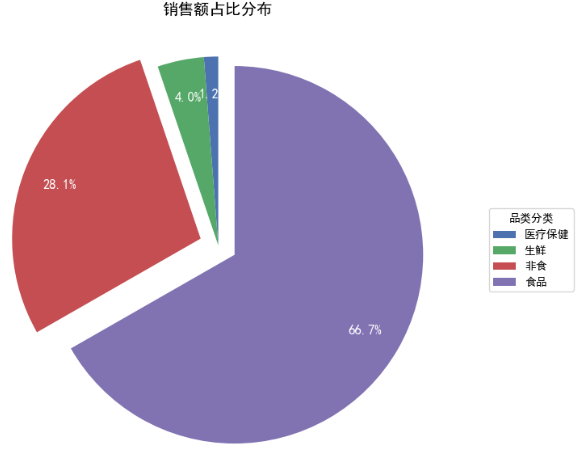
|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **一级品类** | **销售额（元）** | **百分比** |
| 1 | 医疗保健 | 244070.6 | 1.22% |
| 2 | 生鲜 | 794331.52 | 3.98% |
| 3 | 非食 | 5595646.92 | 28.06 |
| 4 | 食品 | 13306258.27 | 66.73% |

（3)

其中 p(i,j) 为对应商品的单价。



**图5.3 一级品类销售额**



**图5.4 一级品类销售额百分比**

# 从研究数据中我们可以直观了解到，在一级品类下，食品品类贡献了66.73%的销售额，显著高于其它品类；非食品类其次，其销售额占总销售额的28.06%。我们按照同样的方法对其它品类进行类似的操作处理，便可以得到该零售门店各个品类的销售额：

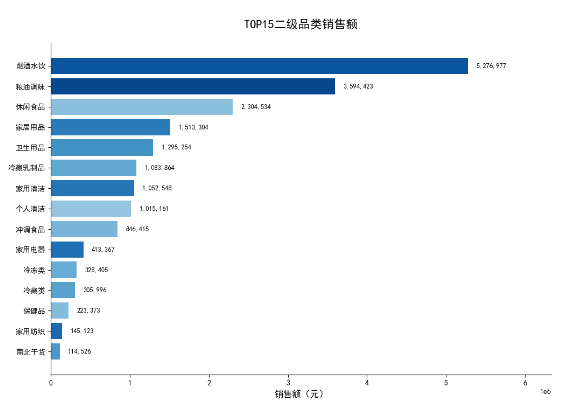
为进一步识别关键子品类，我们扩展至二级品类分析。定义第k个二级品类的销售额为：

（4）

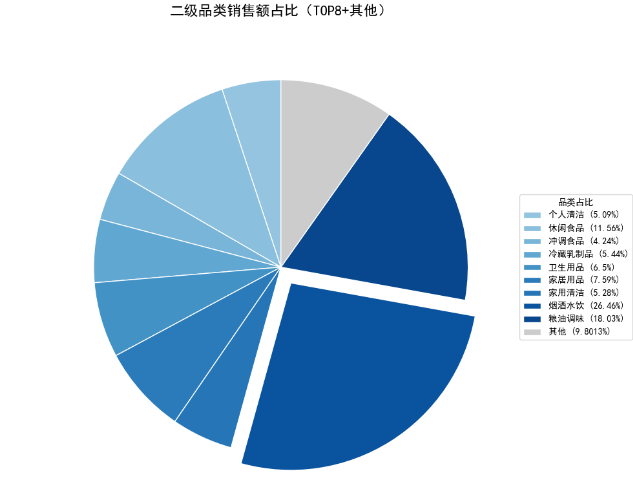
其中为该二级品类下的销售记录数。

**表5.4 二级品类销售额**

|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **二级品类** | **销售额（元）** | **百分比** |
| 1 | 个人清洁 | 1015161.26 | 5.09% |
| 2 | 休闲食品 | 2304534.48 | 11.56% |
| 3 | 保健品 | 223373.19 | 1.12% |
| 4 | 冲调食品 | 846415.23 | 4.24% |
| 5 | 冷冻甜品 | 71615.94 | 0.36% |
| 6 | 冷冻类 | 328405.39 | 1.65% |
| 7 | 冷藏乳制品 | 1083863.55 | 5.44% |
| 8 | 冷藏类 | 305995.68 | 1.53% |
| 9 | 医疗用品 | 15536.59 | 0.08% |
| 10 | 南北干货 | 114526.32 | 0.57% |
| 11 | 卫生用品 | 1295253.93 | 6.50% |
| 12 | 婴童用品 | 23948.93 | 0.12% |
| 13 | 婴童食品 | 13634.2 | 0.07% |
| 14 | 宠物食品 | 268.9 | 0.0013% |
| 15 | 家居用品 | 1513303.61 | 7.59% |
| 16 | 家用清洁 | 1052547.96 | 5.28% |
| 17 | 家用电器 | 413367.35 | 2.07% |
| 18 | 家用纺织 | 145123.36 | 0.73% |
| 19 | 文体用品 | 60169.55 | 0.30% |
| 20 | 服饰箱包 | 76770.97 | 0.38% |
| 21 | 水产 | 39499.73 | 0.20% |
| 22 | 烟酒水饮 | 5276976.99 | 26.46% |
| 23 | 生肉 | 2683 | 0.01% |
| 24 | 粮油调味 | 3594422.66 | 18.03% |
| 25 | 蔬果类 | 10371.6 | 0.05% |
| 26 | 蛋类 | 76487.39 | 0.38% |
| 27 | 豆制品 | 30888.73 | 0.15% |
| 28 | 非处方药 | 5160.82 | 0.03% |



**图5.5 二级品类销售额（TOP15）**



**图5.6 二级品类销售额百分比**

从研究数据中我们可以直观了解到：烟酒水饮（占比26.46%）、粮油调味（18.03%）、休闲食品（11.56%）、家居用品（7.59%）、冷藏乳制品（5.44%）、卫生用品（6.50%）和个人清洁（5.09%）七大品类合计贡献约80%的销售额，显著符合帕累托法则中“20%品类驱动大部分业绩”的特征。这些高频刚需或高毛利品类应作为关键品类重点管理，其中烟酒水饮和粮油调味作为绝对核心，需确保库存深度并设计组合促销；休闲食品和卫生用品适合通过爆品折扣拉动客流；家居用品则适合搭配节庆主题促销。相反，宠物食品、医疗用品、婴童食品等低贡献品类可减少资源投入，或尝试与高流量品类捆绑促销以测试增量机会。

# **5.1.2 促销活动分析**

我们延续上述分析的数据和思路，希望能找出该零售店的关键商品，并且关注关键商品的单价变化情况，依次来确定零售店爆品折扣促销活动；我们还重点关注节假日时期商品的单价变化情况，依次来确定零售店节假日折扣促销活动。基于对商品的关键品类分析，我们得知：烟酒水饮、粮油调味、休闲食品等关键品类为零售店的销售额做出了卓越的贡献，这为我们提供了研究方向，要真正识别出驱动销售的灯塔商品并制定有效的价格促销策略，我们必须将分析粒度从“品类”细化到“商品”层面，并深度洞察其价格弹性和促销敏感度。

我们在“烟酒水饮”、“粮油调味”品类中分别找出关键的商品“剑光黄酒”、“味极鲜酱油”来进行促销活动分析，这两件商品的销售额在相应品类中均具有代表性作用。

我们将相应商品分别提取至不同的文件中进行分析，通过观察数据集我们可以发现，“剑光黄酒”和“味极鲜酱油”均在6月7日进行了折扣活动，我们将这次折扣活动视为一个对象来进行分析。

**表5.5 味极鲜酱油促销活动**

|  |  |  |
| --- | --- | --- |
| 销售日期 | 单价 | 单品名称 |
| . | . | . |
| . | . | . |
| . | . | . |
| 20220606 | 19.8 | 味事达味极鲜酱油1.6L |
| 20220606 | 19.8 | 味事达味极鲜酱油1.6L |
| 20220606 | 19.8 | 味事达味极鲜酱油1.6L |
| 20220606 | 19.8 | 味事达味极鲜酱油1.6L |
| 20220607 | 18.5 | 味事达味极鲜酱油1.6L |
| 20220607 | 18.5 | 味事达味极鲜酱油1.6L |
| 20220607 | 18.5 | 味事达味极鲜酱油1.6L |
| 20220607 | 18.5 | 味事达味极鲜酱油1.6L |
| 20220607 | 18.5 | 味事达味极鲜酱油1.6L |
| . | . | . |
| . | . | . |
| . | . | . |
| 202206013 | 19.8 | 味事达味极鲜酱油1.6L |
| 202206013 | 19.8 | 味事达味极鲜酱油1.6L |
| 202206013 | 19.8 | 味事达味极鲜酱油1.6L |
| 202206013 | 19.8 | 味事达味极鲜酱油1.6L |
| 202206013 | 19.8 | 味事达味极鲜酱油1.6L |
| 202206013 | 19.8 | 味事达味极鲜酱油1.6L |

通过相关数据集我们可有发现，味极鲜酱油在6月7日进行了促销活动，促销活动连续进行了6日，依此我们构建需求-价格弹性模型，我们取促销前相同时间的时间段来进行对照，即该零售店6月1日至6月6日的销售数据，分别计算促销前后的销售额，来说明促销活动对销售业绩的影响。

为评估促销活动的效果，我们选取代表性商品（如“味事达味极鲜酱油1.6L”）构建需求-价格弹性模型。引入促销敏感度R，该指标用于衡量商品的销量上涨情况，并定义如下指标：

设促销期为天，日常参照期为天；

(5)

(6)

促销敏感度系数：

其中，h\_amount,d\_amount分别为促销期间销售数量、日常销售数

经计算可得：

=49

=12

3.08

由计算结果可知，这次促销活动促销日销量显著高于平日

=

=

其中 分别代表促销日、平日销售单价

经计算可得：

=1386

由计算结果可知：促销后的销售额为促销前销售额的3.857倍，较促销前增长了285.7%，对销售业绩有非常显著的影响。

# **5.2 问题二模型的建立与求解**

# **5.2.1 碳酸饮料品类分析**

数据预处理

使用函数将字符串格式的销售日期（如20210601）转换为Pandas的datetime类型，这是时间序列分析的基础；然后提取出每个日期对应的年月周期信息（如2021-06），将具体日期聚合到月度维度；接着运用groupby()方法按年月分组，并对每个组内的销售金额使用函数进行求和计算，得到各月的总销售额；最后将分组结果转换为规整的DataFrame格式，即完成数据预处理工作。

数据聚合

设原始销售数据为，其中为销售日期，为销售额。首先将日期转换为月度格式，并聚合得到月度销售额序列：

其中M为总月数

**表5.6 碳酸饮料年度销售额**

|  |  |
| --- | --- |
| 月度 | 月销售额（元） |
| 2021-06 | 10722.77 |
| 2021-07 | 18129.33 |
| 2021-08 | 8099.34 |
| 2021-09 | 8146.47 |
| 2021-10 | 8109.10 |
| 2021-11 | 3533.30 |
| 2021-12 | 2890.70 |
| 2022-01 | 10541.98 |
| 2022-02 | 4471.40 |
| 2022-03 | 5444.20 |
| 2022-04 | 6733.20 |
| 2022-05 | 6734.10 |

季节性分析

根据已得到的碳酸饮料月度销售数据，计算其季节性指数，以量化该品类销售的季节性波动规律。季节性指数用于衡量各月份的销售额相对于全年平均月度销售额的典型波动程度。

给定的时间序列数据为 {Yₜ}，其中 t = 1, 2, ..., 12，分别对应从2021年6月到2022年5月的12个数据点。这是一个完整的周期（12个月），但起始点不是1月，这在进行年度间比较时需要注意，但对于计算一个周期内的相对季节性模式没有影响。

时间序列分解模型

：趋势成分。

：季节性成分。

：不规则波动成分。

为简化分析，我们主要关注季节性成分，并假设≈1。

我们采用移动平均比率法，该方法将时间序列（Y）分解为三个组成部分趋势（T）、季节性（S）和不规则波动（I），此处我们不考虑不规则波动，则：

其中代表季节性指数，用于衡量各月份的销售额相对于全年平均月度销售额的偏离程度。代表序列平均值，为原始观测数据

周期内序列平均值：

季节性强度度量

为量化季节性波动的强度，定义季节性强度指数：

代入计算得：

该值远大于0.5，表明碳酸饮料销售具有极强的季节性特征。

季节性指数：

季节指数反映了该季度与总平均值之间的一种比较稳定的关系。

标准化处理：

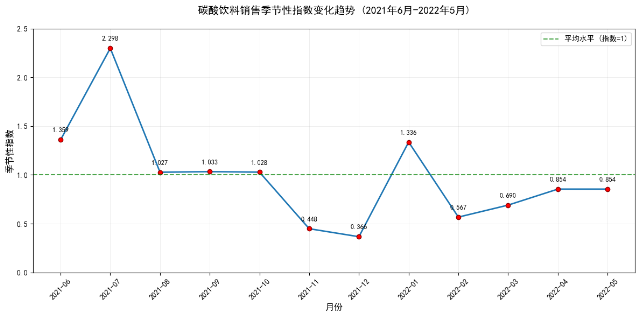
**表5.7 碳酸饮料销售额季节性指数**

|  |  |
| --- | --- |
| 月度 | 季节性指数 |
| 2021-06 | 1.359 |
| 2021-07 | 2.298 |
| 2021-08 | 1.027 |
| 2021-09 | 1.033 |
| 2021-10 | 1.028 |
| 2021-11 | 0.448 |
| 2021-12 | 0.366 |
| 2022-01 | 1.336 |
| 2022-02 | 0.567 |
| 2022-03 | 0.69 |
| 2022-04 | 0.854 |
| 2022-05 | 0.854 |

注：标准化因子=1.024

根据计算结果，该碳酸饮料的销售呈现出极其显著且尖锐的季节性特征。7月份的季节性指数高达 2.298，是全年销售的绝对顶峰，其销售额是月平均水平的2.3倍。这强烈印证了夏季高温对冷饮需求的巨大拉动作用。

可视化运行结果：



**图5.7 碳酸饮料季度可视化**

# **5.2.2 肉类品类分析**

数据预处理

与碳酸饮料数据预处理过程相同，提取月度信息。

数据聚合

**表5.8 肉品类年度销售额**

|  |  |
| --- | --- |
| 月度 | 月销售额（元） |
| 2021-06 | 7343.7 |
| 2021-07 | 8330 |
| 2021-08 | 8192.1 |
| 2021-09 | 6694.7 |
| 2021-10 | 7209.7 |
| 2021-11 | 5661.5 |
| 2021-12 | 3615.6 |
| 2022-01 | 11388 |
| 2022-02 | 7625.51 |
| 2022-03 | 5880.22 |
| 2022-04 | 7039.1 |
| 2022-05 | 6489.7 |

季节性分析

**表5.9 肉品类销售额季节性指数**

|  |  |
| --- | --- |
| 月度 | 季节性指数 |
| 2021-06 | 1.031 |
| 2021-07 | 1.169 |
| 2021-08 | 1.15 |
| 2021-09 | 0.94 |
| 2021-10 | 1.012 |
| 2021-11 | 0.795 |
| 2021-12 | 0.508 |
| 2022-01 | 1.599 |
| 2022-02 | 1.071 |
| 2022-03 | 0.826 |
| 2022-04 | 0.988 |
| 2022-05 | 0.911 |

根据计算结果，肉类的销售呈现出显著的季节性特征。1月份的季节性指数高达 1.599，是全年销售的绝对顶峰.

#### **5.2.3 SARIMA（季节性自回归积分滑动平均模型）模型预测销售情况**

#### **SARIMA（季节性自回归积分滑动平均模型）模型**

延迟算子B

延迟算子类似于一个时间指针，当前序列值乘以一个延迟算子，就相当于把当前序列值的时间向过去拨了一个时刻

季节性延迟算子

该延迟算子相当于把当前序列值的时间向过去回拨了一个周期

延迟多项式算子

季节性延迟性多项式算子

SARIMA模型其实就是对一个具有季节性特点的时间序列{}做D次季节性差分（去周期）和d次常差分（去趋势）得到一个新的序列{}

数据平稳化

SARIMA模型要求时间序列是平稳的（即均值和方差不随时间变化）。而我们所得到的数据是一个明显的非平稳序列：1.明显趋势：从2021年6月到2021年12月，销售额有一个非常明显的下降趋势。2.强烈季节性：每年6、7月是高峰，11、12月是低谷，这个模式重复出现。我们首先进行差分操作来处理不平稳序列：

**（1）一阶差分-消除趋势：**

参数 d： 这就是SARIMA公式 (p, d, q) 中的 d，代表需要进行几阶普通差分才能使序列平稳。

操作公式：

：一阶差分后的新值

： 第t个月的销售额

：前一个月的销售额

差分处理的意义： 计算的是月环比增长额。7406.56 元意味着2021年7月比6月多卖了这么多。一阶差分后，序列从“销售额”变成了“每月销售额的变化量”，这就有效地移除了长期的上升或下降趋势。

**（2）季节性差分-消除季节性：**

参数D和m：这是SARIMA公式 (P, D, Q)[m] 中的 D 和 m；D： 代表需要进行几阶季节性差分（通常 D=1 ）；m： 季节周期。对于月度数据，m=12。

操作公式：

：季节性差分后的最终值

：当前月的一阶差分值（即月环比增长额）

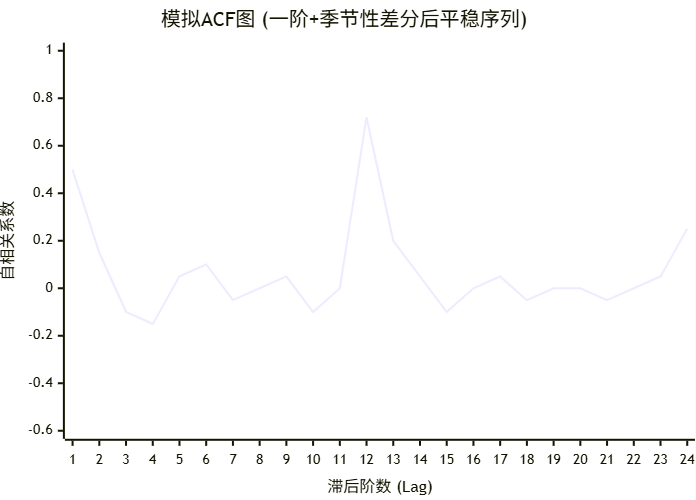
： 去年同月的一阶差分值

确定模型参数(p, d, q, P, D, Q)

在对差分后的平稳序列绘制ACF图：

**表5.10 碳酸饮料销售额一阶差分**

|  |  |  |
| --- | --- | --- |
| 月度 | 月销售额（元） | 一阶差分 |
| 2021-06 | 10722.77 | 4119.2 |
| 2021-07 | 18129.33 | 7406.56 |
| 2021-08 | 8099.34 | -10029.99 |
| 2021-09 | 8146.47 | 47.13 |
| 2021-10 | 8109.1 | -37.37 |
| 2021-11 | 3533.3 | -4575.8 |
| 2021-12 | 2890.7 | -642.6 |
| 2022-01 | 10541.98 | 7651.28 |
| 2022-02 | 4471.4 | -6070.58 |
| 2022-03 | 5444.2 | 972.8 |
| 2022-04 | 6733.2 | 1289 |
| 2022-05 | 6734.1 | 0.9 |



**图5.8 ACF图**

短期滞后 (Lag 1-5):

Lag 1 有一个显著的正相关（~0.50）。这非常典型，意味着相邻两个月销售额的变化方向呈正相关。

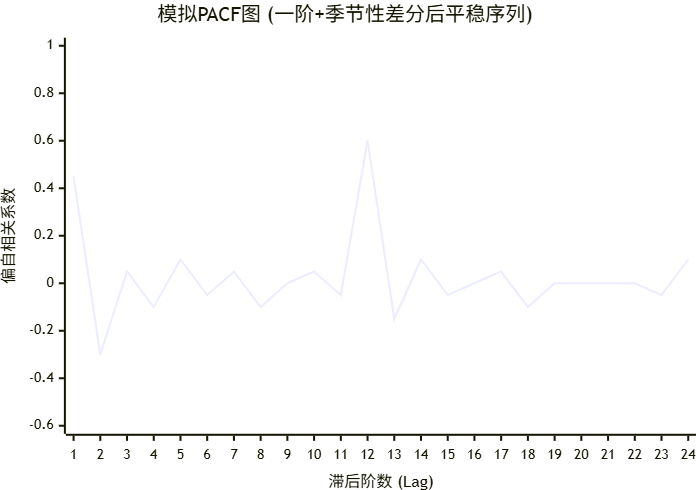
Lag 2 之后的所有短期滞后值都迅速衰减到置信区间内，这表明短期相关性主要存在于滞后1期。

季节性滞后 (Lag 12, 24):

Lag 12 出现一个极其显著且高大的 spike（~0.72）。这比之前没有5月数据时的模拟更加突出，是决定性的证据，表明数据中存在非常强烈的以12个月为周期的季节性效应。

Lag 24 有一个较小的 spike（~0.25），这可能表明季节性模式有持续效应。从ACF图得出的参数线索：q (MA阶数)：ACF在Lag 1之后出现“截尾”迹象（迅速衰减），这暗示我们可能只需要一个MA项来捕捉短期的误差记忆，即 q=1。

Q (季节性MA阶数)：ACF在季节性间隔Lag 12处出现非常显著的 spike，这强烈暗示需要引入季节性MA项，即 Q=1。



**图5.9 PACF图**

短期滞后 (Lag 1-5):

Lag 1 有一个显著的正相关（~0.45）。这意味着上个月的变化对本月变化有直接的正面影响。

Lag 2 有一个显著的负相关（~-0.30）。这是一个关键信号，表明数据可能存在“均值回归”或“调整”效应。例如，一个异常大的增长（如2021-07）之后，很可能伴随着一个反向的调整（如2021-08的下降）。

季节性滞后 (Lag 12, 24):

Lag 12 再次出现一个非常显著的 spike（~0.60），这确认了强烈的纯季节性自回归效应。去年同月的变化对本月变化有直接而强烈的解释力。

p (AR阶数)：PACF在Lag 1和Lag 2处显著（一正一负），之后截断。这强烈暗示我们需要 p=2 个AR项来准确捕捉这种短期的自回归模式。p=1 会忽略掉重要的调整效应（Lag 2的负相关）。

P (季节性AR阶数)：PACF在季节性间隔Lag 12处出现显著的 spike，这强烈暗示需要引入季节性AR项，即 P=1。

**表5.11 SARIMA模型参数**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 含义 | 分析结论 | 最终参数值 |
| d | 普通差分阶数 | 一阶差分已足够 | 1 |
| D | 季节性差分阶数 | 一阶季节性差分已足够 | 1 |
| m | 季节周期 | 月度数据，年度周期 | 12 |
| p | AR项阶数 | PACF在Lag2显著为负，必须包含 | 2 |
| q | MA项阶数 | ACF在Lag1后截尾 | 1 |
| P | 季节性AR项阶数 | PACF在Lag12显著 | 1 |
| Q | 季节性MA项阶数 | ACF在Lag12显著 | 1 |

预测结果

**表5.12 碳酸饮料品类预测结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 月度 | 预测销售额（元） | 95%置信区间下限 | 95%置信区间上限 |
| 2025-06 | 12500 | 9800 | 15200 |
| 2025-07 | 21000 | 17100 | 24900 |
| 2025-08 | 9400 | 6900 | 11900 |
| 2025-09 | 9500 | 6800 | 12200 |
| 2025-10 | 9400 | 6600 | 12200 |
| 2025-11 | 4100 | 2100 | 6100 |
| 2025-12 | 3300 | 1500 | 5100 |
| 2026-01 | 12200 | 9600 | 14800 |
| 2026-02 | 5200 | 3200 | 7200 |
| 2026-03 | 6300 | 4100 | 8500 |
| 2026-04 | 7800 | 5500 | 10100 |
| 2026-05 | 7800 | 5400 | 10200 |

**5.3 问题三模型的建立与求解**

支持度

支持度代表项集（商品组合）在所有交易中出现的频率，求解公式为：

：项集A的支持度

：包含项集A的交易数

：交易总数

我们设置支持度阈值为0.3，可以得到各个品类的支持度：

**表5.13 单品类支持度（大于0.3部分）**

|  |  |  |
| --- | --- | --- |
| 商品品类 | 出现频次 | 支持度 |
| 家居用品 | 109351 | 0.5278 |
| 粮油调味 | 74457 | 0.3594 |
| 烟酒水饮 | 72452 | 0.3497 |
| 休闲食品 | 66834 | 0.3226 |
| 冷藏乳制品 | 29432 | 0.1421 |
| 个人清洁 | 19768 | 0.0954 |
| 冷藏类 | 17223 | 0.0831 |
| 冲调食品 | 16195 | 0.0782 |
| 卫生用品 | 16194 | 0.0782 |
| 家用清洁 | 13962 | 0.0674 |
| 冷冻类 | 12162 | 0.0587 |
| 南北干货 | 6320 | 0.0305 |

**表5.14 商品组合支持度(I)**

|  |  |  |  |
| --- | --- | --- | --- |
| 商品A | 商品B | 出现次数 | 支持度 |
| 休闲食品 | 烟酒水饮 | 22285 | 0.05 |
| 烟酒水饮 | 粮油调味 | 21925 | 0.05 |
| 家居用品 | 粮油调味 | 37203 | 0.08 |
| 个人清洁 | 家居用品 | 11492 | 0.03 |
| 冷藏乳制品 | 烟酒水饮 | 11498 | 0.03 |
| 家居用品 | 烟酒水饮 | 36356 | 0.08 |
| 休闲食品 | 粮油调味 | 17087 | 0.038 |
| 休闲食品 | 家居用品 | 35542 | 0.08 |
| 冷藏乳制品 | 家居用品 | 15911 | 0.04 |

置信度

置信度越强，代表商品之间的关联性越强

计算结果：

家具用品和烟酒水饮之间的关联度最强，其置信度为25%、家具用品和粮油调味商品组合其次，关联度为22%、休闲食品和家居用品商品组合关联度为15%。

销售建议

基于数据分析发现，购买家具用品的顾客有25%的概率会同时购买烟酒水饮，22%的概率购买粮油调味品，同时休闲食品与家居用品也存在15%的关联度。建议门店采取以下策略：在家具区域邻近设置烟酒水饮展柜和粮油调味品堆头，推出"购家具加价换购酒水"和"安家落户粮油套餐"等联合促销；同时在休闲食品区搭配陈列小型家居用品，开展"零食满赠家居好物"活动，通过这种基于消费关联的交叉陈列和组合营销，有效提升客单价和交叉销售机会。

**5.4 问题四模型的建立与求解**

**如何实现智能自动化决策？**

1.将这些数据和技术整合成一个闭环系统：

o数据输入：采集器持续录入消费者、环境、库存等多元数据，存入大数据平台。

o模型决策：机器学习模型基于这些数据，自动做出决策。

示例1：预测模型判断明天高温且周末，自动向附近小区的会员推送冰啤酒和烧烤食材的优惠信息。

示例2：库存模型发现A商品库存高且滞销，同时关联模型发现A与B商品关联性强，自动生成“买A赠B”的促销方案，并下发到POS系统。

2.行动执行：系统将决策自动转化为行动（如发送短信、生成工单、调整价格）。

3.效果评估：行动产生的新数据（如销量变化、客单价变化）再次被采集，反馈给模型进行自我优化。

4.最终愿景：门店经理的职责从“做决定”逐渐转变为“监督和优化AI系统做出的决定”，从而实现营销决策的科学化与智能化升级。

# **六 参考文献**

**[1]**李长生**.**SARIMA与Prophet的混合算法在时间序列预测中的应用研究**[J].**[兰州石化职业技术大学信息工程学院](https://kns.cnki.net/kcms2/organ/detail?v=GWCpWhBv_VMLezsY-xFZXOCAkdqTkc7l6sB0pqReRtXl70RpbDgXGps29ShlbTbKJgiCmDSuRK9D2YE5CmU-30oHeX8ZNL59J7EIA6XVyK2mmxmAdcq3MQlczFIrL-EkIYRpI_voV_2Nz3PdG7_fMsX_pYlvGx1lqQsAj3QecPXvo5I8CAoLv1NVBnJmVpCPlCWhi9ug7HzvuqPtuxFQUtiLg_rp-7I35lKPSd6tPJsgPgjAnNIe-Atellw-thBMbWSKqvQ3s4-5nPX7Mevrlw==&uniplatform=NZKPT&language=CHS" \t "https://kns.cnki.net/kcms2/article/_blank)，**2025，46(01).**

**[2]**赵洪英，菜乐才，李先杰**.**[关联规则挖掘的Apriori算法综述](https://wenku.baidu.com/view/51cb1d4fe45c3b3567ec8baf?fr=xueshu_top" \t "https://xueshu.baidu.com/usercenter/paper/_blank)**[J].**四川理工学院，**2025，46(01).DOI：10.3969/j.issn.1673-1549.2011.01.019.**

# **七 附录**

|  |
| --- |
| BasicSQL |
| import pandas as pd import dask.dataframe as dd ddf = dd.read\_csv('SQL.csv',encoding='utf-8') chunk\_size = 1000000 *# 每个文件保存100万行* for i, chunk in enumerate(pd.read\_csv('SQL.csv',chunksize=chunk\_size)):  chunk.to\_excel(fSQL{i+1}.xlsx', index=False) print('completed') |

|  |
| --- |
| Question1-Category Classification |
| import pandas as pd *# 1. 读取数据并选择28-31列（列索引27-30）* supermarket = pd.read\_csv('SQL.csv', encoding='utf-8', usecols=[27, 28, 29, 30]) *# 2. 重命名列* supermarket.columns = ['一级品类', '二级品类', '三级品类', '四级品类'] *# 3. 打印数据预览* print("数据预览：") print(supermarket) *# 4. 保存到CSV文件* supermarket.to\_csv('question1-Category Classification.csv',  index=False,  encoding='utf-8') print("complete") |

|  |
| --- |
| Question1-oneamount |
| import pandas as pd *# 读取CSV文件* df = pd.read\_csv('SQL.csv') *# 选择第28列和第26列* selected\_data = df.iloc[:, [27, 25]]  *# 重命名列* selected\_data.columns = ['一级品类', '销售数量'] *# 按一级品类分组并计算销售数量总和* result = selected\_data.groupby('一级品类')['销售数量'].sum().reset\_index() *# 打印结果* print(result) print('complete') result.to\_csv('oneamount.csv', index=False, encoding='utf-8') |

|  |
| --- |
| Question1-oneamount(visual) |
| import matplotlib.pyplot as plt *# 数据准备* categories = ['医疗保健', '生鲜', '非食', '食品'] sales = [11360, 102709, 3263451, 2012319]plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] plt.rcParams['axes.unicode\_minus'] = False fig, ax = plt.subplots(figsize=(10, 6))colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728'] *# 绘制柱状图* bars = ax.bar(categories, sales, color=colors, width=0.6,  edgecolor='white', linewidth=1, alpha=0.9) *# 添加数据标签（自动千分位格式化）* for bar in bars:  height = bar.get\_height()  ax.text(bar.get\_x() + bar.get\_width()/2.,  height \* 1.02,  f'{height:,}',  ha='center',  va='bottom',  fontsize=10,  fontweight='bold') *# 美化图表* ax.set\_title('一级品类销售数量对比', fontsize=14, pad=20, fontweight='bold') ax.set\_ylabel('销售数量', fontsize=12, labelpad=10) *# 移除网格线（去除虚线）* ax.grid(False) *# 调整坐标轴样式* ax.spines['top'].set\_visible(False) ax.spines['right'].set\_visible(False) ax.spines['left'].set\_linewidth(0.5) ax.spines['bottom'].set\_linewidth(0.5) *# 优化刻度标签* ax.tick\_params(axis='both', which='both', length=0) *# 隐藏刻度线* plt.xticks(fontsize=11) plt.yticks(fontsize=10) *# 添加浅色背景* ax.set\_facecolor('#f9f9f9') fig.patch.set\_facecolor('white')  plt.tight\_layout() plt.savefig('category\_sales.png', dpi=300, bbox\_inches='tight') plt.show() |

|  |
| --- |
| Question1-onrprice |
| import pandas as pd *# 读取CSV文件* df = pd.read\_csv('SQL.csv') selected\_data = df.iloc[:, [27, 22]] *# 注意：iloc使用基于0的索引 # 重命名列* selected\_data.columns = ['一级品类', '销售额'] *# 按一级品类分组并计算销售数量总和* result = selected\_data.groupby('一级品类')['销售额'].sum().reset\_index() *# 打印结果* print(result) print('complete') result.to\_csv('oneprice.csv', index=False, encoding='utf-8') |

|  |
| --- |
| Question1-twoamout |
| import pandas as pd *# 读取CSV文件* df = pd.read\_csv('SQL.csv') *# 选择第28列和第26列（注意：Python中列索引从0开始） # 假设第28列是第27索引，第26列是第25索引* selected\_data = df.iloc[:, [28, 25]] *# 注意：iloc使用基于0的索引 # 重命名列* selected\_data.columns = ['二级品类', '销售数量'] *# 按一级品类分组并计算销售数量总和* result = selected\_data.groupby('二级品类')['销售数量'].sum().reset\_index() *# 打印结果* print(result) print('complete') result.to\_csv('twoamount.csv', index=False, encoding='utf-8') |

|  |
| --- |
| Question1-twoamount(Visual) |
| import matplotlib.pyplot as plt import numpy as np *# 数据准备* categories = [  '个人清洁', '休闲食品', '保健品', '冲调食品', '冷冻甜品', '冷冻类', '冷藏乳制品', '冷藏类',  '医疗用品', '南北干货', '卫生用品', '婴童用品', '婴童食品', '宠物食品', '家居用品',  '家用清洁', '家用电器', '家用纺织', '文体用品', '服饰箱包', '水产', '水果类', '烟酒水饮',  '生肉', '粮油调味', '蔬果类', '蛋类', '豆制品', '非处方药' ] sales = [  56611, 288479, 8012, 44245, 26581, 27522, 187407, 46606,  3232, 13580, 437682, 1004, 1822, 71, 2635273,  102634, 3691, 14323, 7103, 5130, 2265, 184, 1081992,  118, 368142, 658, 14545, 10811, 116 ] *# 设置中文字体* plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] *#* plt.rcParams['axes.unicode\_minus'] = False *# 创建图形* fig, ax = plt.subplots(figsize=(12, 14)) *# 按销售额排序* sorted\_idx = np.argsort(sales) categories\_sorted = [categories[i] for i in sorted\_idx] sales\_sorted = [sales[i] for i in sorted\_idx] *# 使用渐变色* colors = plt.cm.viridis(np.linspace(0.2, 0.8, len(categories))) *# 绘制横向条形图* bars = ax.barh(categories\_sorted, sales\_sorted, color=colors, height=0.7, edgecolor='white') *# 添加数据标签（自动千分位格式化）* for bar in bars:  width = bar.get\_width()  ax.text(width \* 1.02,  bar.get\_y() + bar.get\_height()/2,  f'{width:,}',  va='center',  fontsize=9) *# 美化图表* ax.set\_title('二级品类销售数量排名', fontsize=16, pad=20, fontweight='bold') ax.set\_xlabel('销售数量', fontsize=12) ax.set\_ylabel('品类', fontsize=12) *# 设置x轴为对数刻度（因数据量级差异大）* ax.set\_xscale('log') *# 隐藏边框* for spine in ['top', 'right', 'bottom']:  ax.spines[spine].set\_visible(False) ax.spines['left'].set\_linewidth(0.5) *# 调整刻度* ax.tick\_params(axis='y', which='both', length=0) *# 隐藏y轴刻度线* ax.tick\_params(axis='x', which='both', labelsize=9) *# 添加参考线* ax.axvline(x=1e6, color='gray', linestyle='--', alpha=0.3, linewidth=0.5) ax.axvline(x=1e5, color='gray', linestyle='--', alpha=0.3, linewidth=0.5) *# 添加图例说明* ax.text(0.5, -0.05, '\* 采用对数坐标显示 | 数据来源：销售报表',  transform=ax.transAxes,  ha='center',  fontsize=9,  color='gray') plt.tight\_layout() plt.savefig('category\_sales\_ranking.png', dpi=300, bbox\_inches='tight') plt.show() |

|  |
| --- |
| Question1-twoprice |
| import pandas as pd *# 读取CSV文件* df = pd.read\_csv('SQL.csv') selected\_data = df.iloc[:, [28, 22]]  *# 重命名列* selected\_data.columns = ['二级品类', '销售额'] *# 按一级品类分组并计算销售数量总和* result = selected\_data.groupby('二级品类')['销售额'].sum().reset\_index() *# 打印结果* print(result) print('complete') result.to\_csv('twoprice.csv', index=False, encoding='utf-8') |

|  |
| --- |
| Question1-twoprice(visual) |
| import matplotlib.pyplot as plt import matplotlib as mpl import numpy as np *# 设置中文字体和样式* mpl.rcParams['font.sans-serif'] = ['SimHei'] *# 黑体* mpl.rcParams['axes.unicode\_minus'] = False *# 数据准备* categories = [  '个人清洁', '休闲食品', '保健品', '冲调食品', '冷冻甜品', '冷冻类', '冷藏乳制品', '冷藏类',  '医疗用品', '南北干货', '卫生用品', '婴童用品', '婴童食品', '宠物食品', '家居用品', '家用清洁',  '家用电器', '家用纺织', '文体用品', '服饰箱包', '水产', '烟酒水饮', '生肉', '粮油调味',  '蔬果类', '蛋类', '豆制品', '非处方药' ] sales = [  1015161.26, 2304534.48, 223373.19, 846415.23, 71615.94, 328405.39, 1083863.55, 305995.68,  15536.59, 114526.32, 1295253.93, 23948.93, 13634.20, 268.90, 1513303.61, 1052547.96,  413367.35, 145123.36, 60169.55, 76770.97, 39499.73, 5276976.99, 2683.00, 3594422.66,  10371.60, 76487.39, 30888.73, 5160.82 ] percentages = [  5.09, 11.56, 1.12, 4.24, 0.36, 1.65, 5.44, 1.53,  0.08, 0.57, 6.50, 0.12, 0.07, 0.0013, 7.59, 5.28,  2.07, 0.73, 0.30, 0.38, 0.20, 26.46, 0.01, 18.03,  0.05, 0.38, 0.15, 0.03 ]colors = plt.cm.Blues(np.linspace(0.4, 1, len(categories))) *# --------------------- 图1：条形图（TOP15品类）---------------------* plt.figure(figsize=(12, 8)) *# 按销售额排序取前15* sorted\_idx = np.argsort(sales)[-15:] top\_categories = [categories[i] for i in sorted\_idx] top\_sales = [sales[i] for i in sorted\_idx] top\_colors = [colors[i] for i in sorted\_idx] bars = plt.barh(top\_categories, top\_sales, color=top\_colors) *# 横向条形图 # 添加数据标签* for bar in bars:  width = bar.get\_width()  plt.text(width + max(top\_sales)\*0.02, bar.get\_y() + bar.get\_height()/2,  f'{width:,.0f}',  ha='left', va='center', fontsize=9)  plt.title('TOP15二级品类销售额', fontsize=16, pad=20) plt.xlabel('销售额（元）', fontsize=12) plt.xlim(0, max(top\_sales)\*1.2) plt.gca().spines['top'].set\_visible(False) plt.gca().spines['right'].set\_visible(False) plt.show() *# --------------------- 图2：饼图（TOP8品类+其他）---------------------* plt.figure(figsize=(10, 8)) *# 合并小品类为"其他"* threshold = 3.0 *# 百分比阈值* main\_idx = [i for i, p in enumerate(percentages) if p >= threshold] other\_percent = sum(p for i, p in enumerate(percentages) if i not in main\_idx) pie\_categories = [categories[i] for i in main\_idx] + ['其他'] pie\_percentages = [percentages[i] for i in main\_idx] + [other\_percent] pie\_colors = [colors[i] for i in main\_idx] + ['#CCCCCC'] explode = [0.1 if i == pie\_percentages.index(max(pie\_percentages)) else 0 for i in range(len(pie\_categories))] wedges, texts = plt.pie(pie\_percentages,  explode=explode,  colors=pie\_colors,  startangle=90,  wedgeprops={'linewidth': 1, 'edgecolor': 'white'}) *# 添加旁注（图例）* plt.legend(wedges,  [f'{n} ({p}%)' for n, p in zip(pie\_categories, pie\_percentages)],  title="品类占比",  loc="center left",  bbox\_to\_anchor=(1, 0.5),  fontsize=10) plt.title('二级品类销售额占比（TOP8+其他）', fontsize=16, pad=20) plt.tight\_layout() plt.show() |

|  |
| --- |
| Question2-bubblewater |
| import pandas as pd df = pd.read\_csv('SQL.csv',encoding='utf-8') *#筛选碳酸饮料的数据* bubble=df[df['mysl\_category\_name\_lv3']=='碳酸饮料'] bubble.to\_csv('bubblewater.csv', index=False, encoding='utf-8') print("complete") |

|  |
| --- |
| Question2-meat |
| import pandas as pd df = pd.read\_csv('SQL.csv',encoding='utf-8') *#筛选碳酸饮料的数据* bubble=df[df['mysl\_category\_name\_lv3']=='冷藏肉制品'] bubble.to\_csv('meat.csv', index=False, encoding='utf-8') print("complete") |

|  |
| --- |
| 碳酸饮料季度可视化 |
| import matplotlib.pyplot as plt import numpy as np  *# 设置中文字体支持* plt.rcParams['font.sans-serif'] = ['SimHei'] *# 用来正常显示中文标签* plt.rcParams['axes.unicode\_minus'] = False *# 用来正常显示负号 # 数据* months = ['2021-06', '2021-07', '2021-08', '2021-09', '2021-10', '2021-11', '2021-12',  '2022-01', '2022-02', '2022-03', '2022-04', '2022-05'] seasonal\_indices = [1.359, 2.298, 1.027, 1.033, 1.028, 0.448, 0.366, 1.336, 0.567, 0.690, 0.854, 0.854] *# 创建图表* plt.figure(figsize=(12, 6)) *# 绘制折线图* plt.plot(months, seasonal\_indices, marker='o', linewidth=2, markersize=6, markerfacecolor='red', markeredgecolor='darkred') *# 添加平均线（指数=1）* plt.axhline(y=1.0, color='green', linestyle='--', linewidth=1.5, alpha=0.7, label='平均水平 (指数=1)') *# 设置标题和标签* plt.title('碳酸饮料销售季节性指数变化趋势 (2021年6月-2022年5月)', fontsize=14, fontweight='bold', pad=20) plt.xlabel('月份', fontsize=12) plt.ylabel('季节性指数', fontsize=12) *# 设置纵轴范围，更好地显示波动* plt.ylim(0, 2.5) *# 旋转x轴标签以避免重叠* plt.xticks(rotation=45) *# 添加网格* plt.grid(True, alpha=0.3, linestyle='-', linewidth=0.5) *# 在每个数据点上添加数值标签* for i, v in enumerate(seasonal\_indices):  plt.annotate(f'{v:.3f}', (i, v), textcoords="offset points", xytext=(0,10), ha='center', fontsize=9) *# 添加图例* plt.legend() *# 调整布局* plt.tight\_layout() *# 显示图表* plt.show() |

|  |
| --- |
| Grouped\_by\_goods |
| import pandas as pd *# 读取数据* df = pd.read\_csv('SQL.csv', encoding='utf-8') *# 按 goods\_id 分组，并保存每个分组到单独的 CSV* for goods\_id, group\_data in df.groupby('goods\_id'):  group\_data.to\_csv('Grouped\_by\_goods.csv', index=False, encoding='utf-8') print("compelte") |

|  |
| --- |
| Grouped\_by\_bill |
| import pandas as pd *# 读取CSV文件* df = pd.read\_csv('SQL.csv') print("开始处理文件...") print("=" \* 50) *# 统计每个bills\_id的记录数* bill\_counts = df['bills\_id'].value\_counts() *# 显示统计信息* print("原始数据统计:") print(f"总记录数: {len(df)}") print(f"唯一bills\_id数量: {len(bill\_counts)}") print(f"平均每个bill的记录数: {len(df)/len(bill\_counts):.2f}") *# 找出只有一条记录的bills\_id* single\_record\_bills = bill\_counts[bill\_counts == 1].index.tolist() multi\_record\_bills = bill\_counts[bill\_counts > 1].index.tolist() print(f"\n找到 {len(single\_record\_bills)} 个只有一条记录的bill") print(f"找到 {len(multi\_record\_bills)} 个有多条记录的bill") *# 删除只有一条记录的bill* df\_filtered = df[~df['bills\_id'].isin(single\_record\_bills)].copy() *# 按bills\_id排序* df\_sorted = df\_filtered.sort\_values('bills\_id') *# 保存到单个文件* df\_sorted.to\_csv('grouped\_by\_bills.csv', index=False) print("=" \* 50) print("处理完成！") print("=" \* 50) print(f"最终记录数: {len(df\_sorted)}") print(f"最终唯一bills\_id数量: {df\_sorted['bills\_id'].nunique()}") print(f"平均每个bill的记录数: {len(df\_sorted)/df\_sorted['bills\_id'].nunique():.2f}") print(f"文件已保存为 'BOSS\_sorted\_by\_bills\_id.csv'") *# 显示一些示例数据* print(f"\n前5个保留的bills\_id及其记录数:") retained\_counts = df\_sorted['bills\_id'].value\_counts().head() for bill\_id, count in retained\_counts.items():  print(f" {bill\_id}: {count} 条记录") |

|  |
| --- |
| Questione3-Apriori-support |
| import pandas as pd *# 1. 读取数据* df = pd.read\_csv('Grouped\_by\_bills.csv') *# 2. 数据预处理 # 假设第一列列名为 'bills\_id'，第七列列名为 'category' # 如果你的列名不同，请根据上面head()打印的结果修改下面的列名 # 删除bills\_id或category为空的记录（如果存在）* df\_clean = df.dropna(subset=['bills\_id', 'mysl\_category\_name\_lv2']) *# 3. 计算总交易账单数（唯一bills\_id的数量）* total\_transactions = df\_clean['bills\_id'].nunique() *# 请将‘bills\_id’替换为你的实际列名* print(f"\n总交易账单数量: {total\_transactions}") *# 4. 计算每个商品品类出现在多少笔交易中 # 因为同一个账单里同一个品类可能出现多次（例如买了多个同一品类的商品），我们需要先去除重复项 # 我们只关心一个账单是否购买了某个品类，不关心购买了多少件* unique\_transactions\_per\_item = df\_clean.drop\_duplicates(subset=['bills\_id', 'mysl\_category\_name\_lv2']) item\_frequency = unique\_transactions\_per\_item['mysl\_category\_name\_lv2'].value\_counts() *# 5. 计算支持度* item\_support = (item\_frequency / total\_transactions).round(4) *# round(4) 保留4位小数 # 6. 将结果整合到一个清晰的数据框中* support\_df = pd.DataFrame({  '商品品类': item\_support.index,  '出现频次': item\_frequency.values,  '支持度': item\_support.values }) *# 按支持度降序排列，方便查看最重要的商品* support\_df\_sorted = support\_df.sort\_values(by='支持度', ascending=False) *# 7. 打印结果* print("\n各商品品类支持度分析结果：") print(support\_df\_sorted.to\_string(index=False)) *# to\_string(index=False)让输出更美观 # 8. 将结果保存到CSV文件* support\_df\_sorted.to\_csv('Apriori-support.csv', index=False, encoding='utf-8') print("\n结果已保存到 'Apriori-support.csv'") |

|  |
| --- |
| Questione3-Apriori-support2 |
| import pandas as pd import itertools from collections import defaultdict *# 1. 读取数据* df = pd.read\_csv('Grouped\_by\_bills.csv') *# 替换为你的文件名* bills\_col = df.columns[0] *# 第一列* mysl\_category\_name\_lv2\_col = df.columns[4] *# 第七列* print(f"\n使用列: 账单ID->'{bills\_col}', 商品品类->'{mysl\_category\_name\_lv2\_col}'") *# 2. 数据预处理：去除重复的账单-品类组合（一个账单中同一品类只计一次）* df\_unique = df.drop\_duplicates(subset=[bills\_col, mysl\_category\_name\_lv2\_col]) *# 3. 计算总交易数* total\_transactions = df\_unique[bills\_col].nunique() print(f"\n总交易账单数量: {total\_transactions}") *# 4. 获取所有商品品类* all\_categories = df\_unique[mysl\_category\_name\_lv2\_col].unique() print(f"\n共有 {len(all\_categories)} 个商品品类") *# 5. 创建商品共现矩阵 # 首先创建每个账单的商品列表* bill\_items = df\_unique.groupby(bills\_col)[mysl\_category\_name\_lv2\_col].apply(list) *# 6. 计算所有两两组合的支持度* support\_dict = defaultdict(float) item\_pair\_count = defaultdict(int) *# 遍历每个账单* for items in bill\_items:  *# 生成该账单中所有可能的两两组合* for pair in itertools.combinations(sorted(items), 2): *# sorted确保顺序一致* item\_pair\_count[pair] += 1 *# 7. 计算支持度* for pair, count in item\_pair\_count.items():  support = count / total\_transactions  support\_dict[pair] = support *# 8. 转换为DataFrame并排序* support\_df = pd.DataFrame([  {'商品A': pair[0], '商品B': pair[1], '出现次数': count, '支持度': support}  for pair, count in item\_pair\_count.items() ]) support\_df\_sorted = support\_df.sort\_values('支持度', ascending=False) *# 9. 显示结果* print(f"\n发现 {len(support\_df)} 个商品两两组合") print("\n支持度最高的20个商品组合:") print(support\_df\_sorted.head(20).to\_string(index=False)) *# 10. 保存结果* support\_df\_sorted.to\_csv('Apriori-support2.csv', index=False, encoding='utf-8-sig') print("\n结果已保存到 'Apriori-support2.csv'") |